

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

THIS PAGE BLANK (USPTO)

REC'D 10 DEC 1999

WIPO PCT

PCT/JP99/05903

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

26.10.99

09/582290

△大U

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

1998年10月27日

出 願 番 号

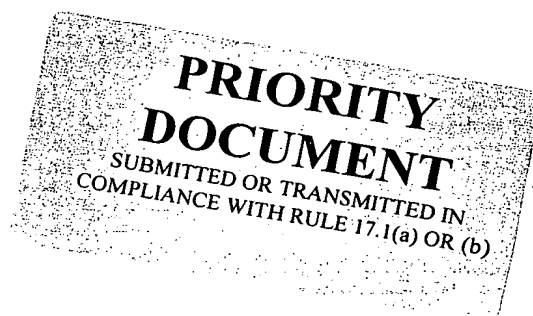
Application Number:

平成10年特許願第321540号

出 願 人

Applicant (s):

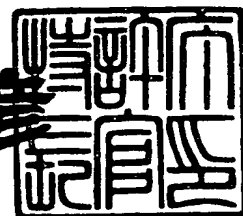
セイコーエプソン株式会社



1999年11月26日

特 許 庁 長 官
Commissioner,
Patent Office

近 藤 隆 彦



出証番号 出証特平11-3081707

【書類名】 特許願

【整理番号】 EP137101

【提出日】 平成10年10月27日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 13/12

【発明の名称】 データ転送制御装置及び電子機器

【請求項の数】 10

【発明者】

 【住所又は居所】 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

 【氏名】 石田 卓也

【発明者】

 【住所又は居所】 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

 【氏名】 神原 義幸

【発明者】

 【住所又は居所】 長野県諏訪市大和3丁目3番5号 セイコーエプソン株式会社内

 【氏名】 和田 文利

【特許出願人】

 【識別番号】 000002369

 【氏名又は名称】 セイコーエプソン株式会社

【代理人】

 【識別番号】 100090479

 【弁理士】

 【氏名又は名称】 井上 一

 【電話番号】 03-5397-0891

【代理人】

 【識別番号】 100090387

【弁理士】

【氏名又は名称】 布施 行夫

【電話番号】 03-5397-0891

【代理人】

【識別番号】 100090398

【弁理士】

【氏名又は名称】 大淵 美千栄

【電話番号】 03-5397-0891

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9402500

【書類名】 明細書

【発明の名称】 データ転送制御装置及び電子機器

【特許請求の範囲】

【請求項1】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、

各ノードから転送されてくる一連の自己識別パケットを受け、上層が使用する制御情報と各ノードから転送されてきた一連の自己識別パケットの並びからなるデータとによりフレームが構成されるパケットに、転送されてきた一連の自己識別パケットを整形するパケット整形手段と、

整形されたパケットを記憶手段に書き込む書き込み手段とを含むことを特徴とするデータ転送制御装置。

【請求項2】 請求項1において、

前記パケット整形手段が、

自己識別パケットがエラーチェック情報を含む場合に、自己識別パケットの中の該エラーチェック情報を削除し、該エラーチェック情報が削除された自己識別パケットの並びからなるデータと前記制御情報とによりフレームが構成されるパケットに、転送されてきた一連の自己識別パケットを整形することを特徴とするデータ転送制御装置。

【請求項3】 請求項2において、

前記パケット整形手段が、

各ノードから転送されてくる自己識別パケットにエラーがあるか否かを前記エラーチェック情報に基づき判断し、自己識別パケットにエラーがあるか否かを示すステータス情報をパケットの制御情報に付加することを特徴とするデータ転送制御装置。

【請求項4】 請求項1乃至3のいずれかにおいて、

前記パケット整形手段が、

自己識別期間に受信したパケットか否かを示すステータス情報をパケットの制御情報に付加することを特徴とするデータ転送制御装置。

【請求項 5】 請求項 1 乃至 4 のいずれかにおいて、

前記記憶手段がランダムアクセス可能であると共に制御情報領域とデータ領域とに分離されている場合において、パケットの制御情報を前記記憶手段の前記制御情報領域に書き込み、パケットのデータを前記記憶手段の前記データ領域に書き込むパケット分離手段と、

前記データ領域に書き込まれるデータのアドレスを指すデータポインタを前記制御情報に付加する手段とを含むことを特徴とするデータ転送制御装置。

【請求項 6】 請求項 5 において、

前記パケット分離手段が、

一連の自己識別パケットを整形することで得られたパケットのデータを、前記記憶手段の自己識別パケット専用設けられた領域に書き込むことを特徴とするデータ転送制御装置。

【請求項 7】 請求項 1 乃至 6 のいずれかにおいて、

下層から送られてくるステータス情報に基づいて自己識別期間中か否かを検出する手段を含み、

前記パケット整形手段が、

自己識別期間中に転送されてきた一連のパケットを自己識別パケットと見なし、該一連の自己識別パケットを整形することを特徴とするデータ転送制御装置。

【請求項 8】 請求項 1 乃至 7 のいずれかにおいて、

IEEE 1394 の規格に準拠したデータ転送を行うことを特徴とするデータ転送制御装置。

【請求項 9】 請求項 1 乃至 8 のいずれかのデータ転送制御装置と、

前記データ転送制御装置及びバスを介して他のノードから受信したデータに所与の処理を施す装置と、

処理が施されたデータを出力又は記憶するための装置とを含むことを特徴とする電子機器。

【請求項 10】 請求項 1 乃至 8 のいずれかのデータ転送制御装置と、

前記データ転送制御装置及びバスを介して他のノードに送信するデータに所与

の処理を施す装置と、

処理が施されるデータを取り込むための装置とを含むことを特徴とする電子機器。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データ転送制御装置及びこれを含む電子機器に関する。

【0002】

【背景技術及び発明が解決しようとする課題】

近年、IEEE1394と呼ばれるインターフェース規格が脚光を浴びている。このIEEE1394は、次世代のマルチメディアにも対応可能な高速シリアルバスインターフェースを規格化したものである。このIEEE1394によれば、動画像などのリアルタイム性が要求されるデータも扱うことができる。また、IEEE1394のバスには、プリンタ、スキャナ、CD-Rドライブ、ハードディスクドライブなどのコンピュータの周辺機器のみならず、ビデオカメラ、VTR、TVなどの家庭用電化製品も接続できる。このため、電子機器のデジタル化を飛躍的に促進できるものとして期待されている。

【0003】

このようなIEEE1394の概要については、例えば「IEEE1394ハイ・パフォーマンス・シリアルバスの概要」(Interface Apr. 1996の1~10頁)、「PC周辺機器用バス規格群総覧」(Interface Jan. 1997の106頁~116頁)、「IEEE1394-1995(FireWire)のリアルタイム転送モードとマルチメディア対応プロトコル」(Interface Jan. 1997の136~146頁)に開示されている。また、IEEE1394に準拠したデータ転送制御装置としては、テキサス・インスツルメンツ社製のTSB12LV31などが知られている。

【0004】

さて、このIEEE1394では、バスリセットの後にツリー識別が行われ、その後に自己識別が行われる。そして、この自己識別の際には、各ノードが自身

のセルフIDパケットを全てのノードにブロードキャストする。そして、セルフIDパケットを受け取った各ノードは、これらのセルフIDパケットを、自身が有するメモリ上に必要に応じて保持しておく。

【0005】

しかしながら、このセルフIDパケットの個数は、ノード数に応じて増えるため、非常に数の多いものとなる。また、この多数のセルフIDパケットを、他のLINKパケットと区別してファームウェアなどの上層に対して正しくインターフェースするためには、少なくとも1クワドレットのヘッダを付加する必要がある。従って、この多数のセルフIDパケットの存在に起因して、各ノードが有するメモリの空き容量が圧迫されてしまい、各ノードが有するリソースの有効利用を図れないという問題が生じる。

【0006】

また、IEEE 1394に準拠したデータ転送制御装置には、システム全体の実転送速度を向上させるため、CPU上で動作するファームウェアやアプリケーションソフトの処理のオーバーヘッドを、なるべく軽減することが望まれる。

【0007】

本発明は、以上のような技術的課題に鑑みてなされたものであり、その目的とするところは、各ノードが有するリソースを有効利用できるデータ転送制御装置及びこれが用いられる電子機器を提供することにある。

【0008】

また、本発明の他の目的は、ファームウェアやアプリケーションソフトなどの処理のオーバーヘッドを軽減できるデータ転送制御装置及びこれが用いられる電子機器を提供することにある。

【0009】

【課題を解決するための手段】

上記課題を解決するために本発明は、バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、各ノードから転送されてくる一連の自己識別パケットを受け、上層が使用する制御情報と各ノードから転送されてきた一連の自己識別パケットの並びからなるデータとによりフレームが構成さ

れるパケットに、転送されてきた一連の自己識別パケットを整形するパケット整形手段と、整形されたパケットを記憶手段に書き込む書き込み手段とを含むことを特徴とする。

【0010】

本発明によれば、一連の自己識別パケットを1つにパッケージングし制御情報を付加することでパケット整形が行われ、一連の自己識別パケットが上層にインターフェースされる。従って、各自己識別パケットに制御情報を付加して上層にインターフェースする場合に比べて、整形されたパケットが書き込まれる記憶手段の記憶領域を節約できる。また、一連の自己識別パケットをひとまとめにして取り扱えるため、ファームウェアなどの処理のオーバーヘッドを軽減できる。

【0011】

また本発明は、前記パケット整形手段が、自己識別パケットがエラーチェック情報を含む場合に、自己識別パケットの中の該エラーチェック情報を削除し、該エラーチェック情報が削除された自己識別パケットの並びからなるデータと前記制御情報とによりフレームが構成されるパケットに、転送されてきた一連の自己識別パケットを整形することを特徴とする。このようにすれば、整形されたパケットが書き込まれる記憶手段の記憶領域を更に節約することが可能になる。

【0012】

また本発明は、前記パケット整形手段が、各ノードから転送されてくる自己識別パケットにエラーがあるか否かを前記エラーチェック情報に基づき判断し、自己識別パケットにエラーがあるか否かを示すステータス情報をパケットの制御情報に付加することを特徴とする。このようにすれば、自己識別パケットにエラーがあったことを、簡易に上層に伝えることが可能になる。

【0013】

また本発明は、前記パケット整形手段が、自己識別期間に受信したパケットか否かを示すステータス情報をパケットの制御情報に付加することを特徴とする。このようにすれば、自己識別パケットをパッケージングすることで得られたパケットとそれ以外のパケットとの区別が容易になる。

【0014】

また本発明は、前記記憶手段がランダムアクセス可能であると共に制御情報領域とデータ領域とに分離されている場合において、パケットの制御情報を前記記憶手段の前記制御情報領域に書き込み、パケットのデータを前記記憶手段の前記データ領域に書き込むパケット分離手段と、前記データ領域に書き込まれるデータのアドレスを指すデータポインタを前記制御情報に付加する手段とを含むことを特徴とする。このようにすれば、整形されたパケットの制御情報の部分とデータの部分とを別個に取り扱うことが可能になり、ファームウェアなどの処理のオーバーヘッドを軽減できる。

【0015】

また本発明は、前記パケット分離手段が、一連の自己識別パケットを整形することで得られたパケットのデータを、前記記憶手段の自己識別パケット専用に分けられた領域に書き込むことを特徴とする。このようにすれば、トポロジーマップの作成などの際の処理を簡易化できるようになる。

【0016】

また本発明は、下層から送られてくるステータス情報に基づいて自己識別期間中か否かを検出する手段を含み、前記パケット整形手段が、自己識別期間中に転送されてきた一連のパケットを自己識別パケットと見なして、該一連の自己識別パケットを整形することを特徴とする。このようにすれば、処理対象となるパケットが自己識別パケットか否かを簡易に検出できるようになり、ハードウェアの小規模化を図れる。

【0017】

なお、本発明では、IEEE 1394の規格に準拠したデータ転送を行うことが望ましい。

【0018】

また本発明に係る電子機器は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードから受信したデータに所与の処理を施す装置と、処理が施されたデータを出力又は記憶するための装置とを含むことを特徴とする。また本発明に係る電子機器は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードに送信するデ

ータに所与の処理を施す装置と、処理が施されるデータを取り込むための装置とを含むことを特徴とする。

【0019】

本発明によれば、他のノードから転送されたデータを電子機器において出力したり記憶したりする処理、電子機器において取り込んだデータを他のノードに転送したりする処理を高速化することが可能になる。また、本発明によれば、整形後のパケットを記憶する記憶手段を小規模化できると共に、データ転送を制御するファームウェアなどの処理負担を軽減できるため、電子機器の低コスト化、小規模化などを図ることも可能になる。

【0020】

【発明の実施の形態】

以下、本発明の好適な実施形態について図面を用いて詳細に説明する。

【0021】

1. IEEE1394

まず、IEEE1394の概要について簡単に説明する。

【0022】

1. 1 データ転送速度、接続トポロジー

IEEE1394 (IEEE1394-1995、IEEE1394. A) では100~400Mbpsの高速なデータ転送が可能となっている (IEEE1394. Bでは800~3200Mbps)。また、転送速度が異なるノードをバスに接続することも許される。

【0023】

各ノードはツリー状に接続されており、1つのバスに最大で63個のノードが接続可能になっている。なお、バスブリッジを利用すれば約64000個のノードを接続することも可能である。

【0024】

電源が投入されたり、途中でデバイスの抜き差しが発生すると、バスリセットが発生し、接続トポロジーに関する情報が全てクリアされる。そして、バスリセット後、ツリー識別 (ルートノードの決定)、自己識別が行われる。その後、ア

アイソクロナスリソースマネージャ、サイクルマスタ、バスマネージャ等の管理ノードが決定される。そして、通常の packets 転送が開始される。

【0025】

1. 2 転送方式

IEEE 1394 では、packets の転送方式として、信頼性が要求されるデータの転送に好適な非同期転送と、リアルタイム性が要求される動画像や音声などのデータの転送に好適なアイソクロナス転送が用意されている。

【0026】

図 1 (A) に、非同期サブアクションの例を示す。1つのサブアクションは、調停、packets 転送、acknowledgment からなる。即ち、データ転送に先立って、まず、バスの使用権に関する調停が行われる。そしてソース (転送元) ノードから destination (転送先) ノードに packets が転送される。この packets のヘッダにはソース ID と destination ID が含まれる。destination ノードは、この destination ID を読んで、自ノード宛の packets か否かを判断する。destination ノードは、packets を受け取ると、ソースノードに acknowledgment (ACK) の packets を返す。

【0027】

packets 転送と ACK の間には acknowledgment ギャップが存在する。また、1つのサブアクションと次のサブアクションの間にはサブアクションギャップが存在する。そして、サブアクションギャップに相当する一定のバス・アイドル時間が経過しないと、次のサブアクションの調停を開始できない。これによりサブアクション相互の衝突が回避される。

【0028】

図 1 (B) に、アイソクロナスサブアクションの例を示す。アイソクロナス転送はブロードキャスト (バスに接続される全てのノードに転送) で実行されるため、packets 受信時に ACK は返送されない。また、アイソクロナス転送では、ノード ID ではなくチャンネル番号を使用して packets 転送が行われる。なお、サブアクション間にはアイソクロナスギャップが存在する。

【0029】

図1 (C) に、データ転送時のバスの様子を示す。アイソクロナス転送は、サイクルマスタが一定周期毎にサイクルスタートパケットを発生することで開始する。これにより、1つのチャンネル当たり、 $125\mu\text{s}$ 毎に少なくとも1つのパケットを転送できるようになる。この結果、動画像や音声などのリアルタイム性が要求されるデータの転送が可能になる。

【0030】

非同期転送はアイソクロナス転送の合間に行われる。即ち、アイソクロナス転送の方が非同期転送よりも優先順位が高くなっている。これは、図1 (C) に示すように、アイソクロナスギャップの時間を、非同期転送のサブアクションギャップの時間よりも短くすることで実現される。

【0031】

1. 3 ツリー識別

ツリー識別はバスリセットの後に行われる。ツリー識別によりノード間の親子関係やルートノードが決定される。

【0032】

まず、リーフノード(1つのノードにしか接続されていないノード)が、隣接するノードに、ペアレントノーティファイを送る。例えば図2 (A) のようにノードA、B、C、D、Eが接続されている場合には、ノードAからノードBに、ノードD及びEからノードCに、ペアレントノーティファイ(PN)が送られる。

【0033】

ペアレントノーティファイを受け取ったノードは、送り元のノードを自身の子と認知する。そして、チャイルドノーティファイをそのノードに送る。例えば図2 (A) では、ノードBからノードAに、ノードCからノードD及びEにチャイルドノーティファイ(CN)が送られる。これによってノードB、A間、ノードC、D間、ノードC、E間の親子関係が決定する。

【0034】

ノードB、Cの親子関係は、どちらが先にペアレントノーティファイを送ったかで決定される。例えば図2 (B) のように、ノードCの方が先にペアレントノ

ーティファイを送ると、ノードBが親になりノードCが子になる。

【0035】

ポートの接続先の全てのノードが自身の子となるノードがルートになる。図2(B)では、ノードBがルートになる。なお、IEEE1394では、全てのノードがルートになる可能性がある。

【0036】

1.4 自己識別

ツリー識別の後、自己識別が行われる。自己識別においては、接続トポロジーにおいてルートノードから遠いノードから順にセルフIDパッケージが転送される。

【0037】

より具体的には、例えば図3において、まず、ルートノードBのポート1（番号の小さいポート）に接続されるノードAが、セルフIDパッケージ（自己識別パッケージ）を全てのノードにブロードキャストする。

【0038】

次に、ルートノードBのポート2（番号が大きいポート）に接続されるノードCが選択され、このノードCのポート1（番号の小さいポート）に接続されるノードDがセルフIDパッケージをブロードキャストする。次に、ノードCのポート2（番号の大きいポート）に接続されるノードEがセルフIDパッケージをブロードキャストし、その後、ノードCがブロードキャストする。最後に、ルートであるノードBがセルフIDパッケージをブロードキャストし、自己識別が完了する。

【0039】

セルフIDパッケージには各ノードのIDが含まれる。ブロードキャストを行う時点で他のノードから受け取ったセルフIDパッケージの個数が、この各ノードのIDとなる。例えば図3では、ノードAがブロードキャストを行う時点では、どのノードもセルフIDパッケージを発していないため、ノードAのIDは0になる。ノードAは、このID=0をセルフIDパッケージに含ませてブロードキャストする。また、ノードDがブロードキャストを行う時点では、ノードAのみがセルフIDパッケージを発している。このため、ノードDのIDは1になる。同様に、

ノードE、C、BのIDは、各々、2、3、4になる。

【0040】

図4 (A) にセルフIDパケットのフォーマットを示す。同図に示すようにセルフIDパケットには各ノードの基本情報が含まれる。具体的には、各ノードのID (PHY_ID)、リンク層がアクティブか否か (L)、ギャップカウント (gap_cnt)、転送速度 (sp)、アイソクロナスリソースマネージャになれる能力を有するか否か (C)、電力状態 (pwr)、ポートの状態 (p0、p1、p2) などに関する情報が含まれる。

【0041】

なお、図4 (B) に、ノードのポート数が4個以上の場合に使用されるセルフIDパケット#1、#2、#3のフォーマットを示す。ポート数が4～11個の場合にはセルフIDパケット#0 (図4 (A)) 及び#1が、12～19個の場合にはセルフIDパケット#0、#1及び#2が、20～27個の場合にはセルフIDパケット#0、#1、#2及び#3が使用されることになる。

【0042】

また、図4 (C)、(D) に、セルフIDパケットと同様に、物理層のパケット (PHYパケット) であるリンクオンパケット、PHY構成パケットのフォーマットを示す。

【0043】

1. 5 アイソクロナスリソースマネージャ

アイソクロナスリソースマネージャ (IRM) は以下の管理機能を有する。

【0044】

第1に、アイソクロナス転送に必要な種々のリソースを提供する。例えば、チャンネル番号レジスタや帯域幅レジスタを提供する。第2に、バスマネージャのIDを示すレジスタを提供する。第3に、バスマネージャがない場合に、簡易的なバスマネージャとなる役割を有する。

【0045】

IRMになれる能力を有し (アイソクロナスリソースを管理する能力を有し)、且つ、動作状態になっている (リンク層がアクティブになっている) ノードの

中で（IRMになれる資格を有するノードの中で）、ルートに最も近い（IDが最も大きい）ノードがIRMになる。より、具体的には、図4（A）のセルフIDパケットにおいて、IRMになれる能力を有するか否かを示すC（CONTENDER）ビットと、リンク層がアクティブか否かを示すL（LINK_ACTIVE）ビットが共に1になっているノードの中で、ルートに一番近いノード（PHY_IDが一番大きいノード）がIRMになる。例えば、ルートノードのセルフIDパケットのCビットとLビットが1の場合には、ルートノードがIRMになる。

【0046】

1. 6 サイクルマスタ、バスマネージャ

サイクルマスタは、図1（C）に示すサイクルスタートパケットを送信する役割を有し、ルートノードがサイクルマスタになる。

【0047】

バスマネージャは、トポロジーマップ（各ノードの接続状態）の作成、スピードマップの作成、バスの電力管理、サイクルマスタの決定、ギャップカウントの最適化などの仕事を行う。

【0048】

1. 7 プロトコル構成

図5を用いて、IEEE 1394のプロトコル構成（層構造）について説明する。

【0049】

IEEE 1394のプロトコルは、物理層、リンク層、トランザクション層により構成される。また、シリアルバスマネージメントは、物理層、リンク層、トランザクション層をモニターしたり制御したりするものであり、ノードの制御やバスのリソース管理のための種々の機能を提供する。

【0050】

物理層は、リンク層により使用されるロジカルシンボルを電気信号に変換したり、バスの調停を行ったり、バスの物理的インターフェースを定義する。

【0051】

リンク層は、アドレッシング、データチェック、データフレーミング、サイクル制御などを提供する。

【0052】

トランザクション層は、リード、ライト、ロックなどのトランザクションを行うためのプロトコルを定義する。

【0053】

物理層及びリンク層は、通常、データ転送制御装置（インターフェースチップ）などのハードウェアにより実現される。また、トランザクション層は、CPU上で動作するファームウェアや、ハードウェアにより実現される。

【0054】

2. 全体構成

次に、本実施形態の全体構成について図6を用いて説明する。

【0055】

図6において、PHYインターフェース10は、物理層のプロトコルを実現するPHYチップとのインターフェースを行う回路である。

【0056】

リンクコア20は、リンク層のプロトコルやトランザクション層のプロトコルの一部を実現する回路であり、ノード間でのパケット転送のための各種サービスを提供する。レジスタ22は、これらのプロトコルを実現したりリンクコア20を制御するためのレジスタである。

【0057】

FIFO(ATF)30、FIFO(ITF)32、FIFO(RF)34は、各々、非同期送信用、アイソクロナス送信用、受信用のFIFOであり、例えばレジスタや半導体メモリなどのハードウェアにより構成される。本実施形態では、これらのFIFO30、32、34の段数は非常に少ない。例えば1つのFIFOの段数は、好ましくは3段以下であり、更に好ましくは2段以下となる。

【0058】

DMAC40、42、44は、各々、ATF、ITF、RF用のDMAコントローラである。これらのDMAC40、42、44を用いることで、CPU66

に介入されることなく、RAM 80 とリンクコア 20 との間でのデータ転送が可能になる。なお、レジスタ 46 は、DMAC 40、42、44などを制御するレジスタである。

【0059】

ポートインターフェース 50 は、アプリケーション層のデバイス（例えばプリンタの印字処理を行うデバイス）とのインターフェースを行う回路である。本実施形態では、このポートインターフェース 50 を用いて、例えば 8 ビットのデータ転送が可能になっている。

【0060】

FIFO (PF) 52 は、アプリケーション層のデバイスとの間でのデータ転送のための FIFO であり、DMAC 54 は、PF 用の DMA コントローラである。レジスタ 56 は、ポートインターフェース 50 や DMAC 54 を制御するレジスタである。

【0061】

CPU インターフェース 60 は、データ転送制御装置をコントロールする CPU 66 とのインターフェースを行う回路である。CPU インターフェース 60 は、アドレスデコーダ 62、データ同期化回路 63、割り込みコントローラ 64 を含む。クロック制御回路 68 は、本実施形態で使用されるクロックを制御するものであり、PHY チップから送られてくる SCLK や、マスタークロックである HCLK が入力される。

【0062】

バッファマネージャ 70 は、RAM 80 とのインターフェースを管理する回路である。バッファマネージャ 70 は、バッファマネージャの制御のためのレジスタ 72、RAM 80 へのバス接続を調停する調停回路 74、各種の制御信号を生成するシーケンサ 76 を含む。

【0063】

RAM 80 は、ランダムアクセス可能なパケット記憶手段として機能するものであり、その機能は例えば SRAM、DRAM などにより実現される。そして、本実施形態では、図 7 に示すように、この RAM 80 がヘッダ領域（広義には制

御情報領域)とデータ領域に分離されている。そして、パケットのヘッダ(広義には制御情報)は図7のヘッダ領域に格納され、パケットのデータはデータ領域に格納される。

【0064】

なおRAM80は、本実施形態のデータ転送制御装置に内蔵させることが特に望ましいが、その一部又は全部を外付けにすることも可能である。

【0065】

バス90(或いはバス92、94)は、アプリケーションに接続されるものである(第1のバス)。またバス96(或いはバス98)はデータ転送制御装置をコントロールするためのものであり、データ転送制御装置をコントロールするデバイス(例えばCPU)に電氣的に接続される(第2のバス)。またバス100(或いはバス102、104、105、106、107、108、109)は、物理層のデバイス(例えばPHYチップ)に電氣的に接続されるものである(第3のバス)。また、バス110は、ランダムアクセス可能な記憶手段であるRAMに電氣的に接続されるものである(第4のバス)。

【0066】

バッファマネージャ70の調停回路74は、DMAC40、DMAC42、DMAC44、CPUインターフェース60、DMAC54からのバスアクセス要求の調停を行う。そして、この調停結果に基づいて、各々、バス105、107、109、98、94のいずれかと、RAM80のバス110との間にデータの経路が確立される(第1、第2、第3のバスのいずれかと第4のバスとの間にデータ経路が確立される)。

【0067】

本実施形態の1つの特徴は、ランダムアクセスが可能でありパケットを格納するRAM80を設けると共に、互いに分離されるバス90、96、100と、これらのバスをRAM80のバス110に接続するための調停回路74とを設けた点にある。

【0068】

例えば図8に、本実施形態と構成の異なるデータ転送制御装置の例を示す。こ

のデータ転送制御装置では、リンクコア902は、PHYインターフェース900、バス922を介してPHYチップと接続される。また、リンクコア902は、FIFO904、906、908、CPUインターフェース910、バス920を介してCPU912に接続される。そして、CPU912は、バス924を介して、CPUにローカルなメモリであるRAM914に接続される。

【0069】

なお、FIFO904、906、908は、図6のFIFO30、32、34と異なり、非常に段数の多いものとなる（例えば1つのFIFOが16段程度）。

【0070】

図8の構成のデータ転送制御装置を用いた場合のデータ転送の手法について図9を用いて説明する。PHYチップ930を介して他のノードから送られてきた受信パケットは、バス922、データ転送制御装置932、バス920を介してCPU912が受け取る。そして、CPU912は、受け取った受信パケットをバス924を介してRAM914に書き込む。そして、CPU912は、受信パケットをアプリケーション層が使用できるように加工し、バス926を介してアプリケーション層のデバイス934に転送する。

【0071】

一方、アプリケーション層のデバイス934からのデータを転送する場合には、CPU912は、このデータをRAM914に書き込む。そして、RAM914のデータにヘッダを付加することでIEEE1394に準拠したパケットを生成する。そして生成されたパケットは、データ転送制御装置932、PHYチップ930などを介して他のノードに送信される。

【0072】

しかしながら、このようなデータ転送手法によると、CPU912の処理負荷が非常に重くなる。従って、ノード間を接続するシリアルバスの転送速度が高速になっても、CPU912の処理のオーバーヘッドなどに起因して、システム全体の実転送速度は低くなり、結局、高速なデータ転送を実現できない。

【0073】

このような問題を解決する 1 つの手法として、図 10 に示すように、データ転送制御装置 932 と RAM 914 との間でのデータ転送や、RAM 914 とアプリケーション層のデバイス 934 との間でのデータ転送を、ハードウェア DMA により実現する手法も考えられる。

【0074】

しかしながら、この手法では、CPU バス 928 が、データ転送制御装置 932、RAM 914 間でのデータ転送、RAM 914、CPU 912 間でのデータ転送、RAM 914、アプリケーション層デバイス 934 間でのデータ転送に使用されることになる。従って、システム全体のデータ転送の高速化を図ろうとすると、CPU バス 928 として PCI バスのような高速なバスを使用しなければならなくなり、これは、データ転送制御装置を使用する電子機器の高コスト化を招く。

【0075】

これに対して、本実施形態では図 11 に示すように、データ転送制御装置 120、アプリケーション層デバイス 124 間のバス 90 と、CPU バス 96 と、データ転送制御装置 120、RAM 80 間のバス 110 とが互いに分離されている。従って、CPU バス 96 をデータ転送の制御のみに使用できるようになる。また、バス 90 を占有して、データ転送制御装置 120、アプリケーション層デバイス 124 間でデータ転送を行うことができるようになる。例えば、データ転送制御装置 120 が組み込まれる電子機器がプリンタである場合には、バス 90 を占有して印字データを転送できるようになる。この結果、CPU 66 の処理負荷を軽減でき、システム全体の転送速度を高めることができる。また CPU 66 として安価なものを採用できると共に、CPU バス 96 として高速なバスを使用する必要性がなくなる。このため、電子機器の低コスト化、小規模化を図れるようになる。

【0076】

3. セルフ ID パケットの整形

3.1 本実施形態の特徴

さて、図 12 に示すように、セルフ ID 期間においては、バスに接続される全

てのノードからセルフIDパケットが送られてくる。そして、各セルフIDパケットの第1クワドレットは、ノードIDなどの基本情報が含まれるデータ本体（ボディ）になっており、第2クワドレットは、第1クワドレットの反転であるパリティ（エラーチェック情報）になっている（詳しくは、図4（A）参照）。

【0077】

例えば図8の構成のデータ転送制御装置では、これらの一連のセルフIDパケットを図13（A）に示すように整形する。即ち、物理層のパケットであるセルフIDパケットに、上層（トランザクション層やアプリケーション層）にインターフェースするためのヘッダを付加するというパケット整形が行われる。そして、整形されたパケットは、CPUのローカルメモリであるRAM（図8のRAM 914）に格納される。

【0078】

しかしながら、セルフIDパケットは、ノード数に応じた個数だけ必要であり、バスに63個のノードが接続されると例えば最小で63個のセルフIDパケットを保持する必要がある。即ち、セルフIDパケットの個数は、通常、非常に多い。従って、これらのセルフIDパケットにヘッダを付加し、そのままRAMに格納すると、RAMの空き容量が圧迫されてしまい、各ノードのリソースを有効利用できなくなってしまう。また、それぞれのセルフIDパケットを受け取る毎に、ファームウェアは必要な処理を行わなければならない。従ってセルフIDパケットの個数が増加すると、ファームウェアの処理負担が増加する。更に、セルフID期間が終了した後、即ちバスリセットが発生してから最初のアービトレーション・リセット・ギャップが発生した後には、その受け取ったセルフIDパケットの整形作業が必要になる。従って、セルフIDパケットの個数が増加すると、この整形作業の処理負担も増加する。

【0079】

そこで、本実施形態では図13（B）に示すように、転送されてくる一連のパケットを、ヘッダ（広義には制御情報）と、一連のセルフIDパケットからなるデータとによりフレームが構成されるパケットに整形している。つまり、一連のセルフIDパケットを1つにパッケージングし、このパッケージングされたもの

に1つのヘッダを付加し、上層にインターフェースしている。

【0080】

図13 (A) の比較例では、セルフIDパケットの各々に一対一に対応してヘッダが付加されている。これに対して、本実施形態では、このように全てのセルフIDパケットの各々にヘッダを付加することは無駄であるということに着目し、一連のセルフIDパケットを1つにパッケージングしたものに1つのヘッダを付加している。

【0081】

このようにすることで、本実施形態では、パケット記憶メモリであるRAM (図6のRAM80) の空き容量が圧迫される事態を効果的に解消することに成功している。これにより、RAMの小規模化を図れ、データ転送制御装置や電子機器の低コスト化、小規模化を図れるようになる。

【0082】

また、図13 (A) の比較例では、セルフIDパケットがN個あった場合には、整形後のパケットの個数もN個になる。従って、これらの整形後のパケットをRAMから読み出すためには、N回のパケット読み出しを行う必要がある。

【0083】

これに対して、本実施形態では、セルフIDパケットがN個あった場合にも、整形後のパケットは1個になる。従って、整形後のパケットをRAMから読み出すためには、1回のパケット読み出しを行えば済むようになる。従って、ファームウェアなどの処理負担を格段に軽減できる。このため、安価なCPUを採用することも可能となり、データ転送制御装置や電子機器の低コスト化、小規模化を図れるようになる。

【0084】

また、本実施形態では、一連のセルフIDパケットが1つにまとめられるため、図13 (A) の比較例に比べて、パケットの取り扱いが簡易になるという利点もある。

【0085】

また、本実施形態では、図14 (A) に示すように、セルフIDパケットの第

2クワドレットのパリティを削除し、このパリティが削除されたセルフIDパケット（セルフIDパケットのデータ本体）の並びにより、整形後のパケットのデータ部分を構成するようにしている。このようにすることで、整形後のパケットのサイズを、パリティを削除しない場合に比べて半分程度にすることができる。この結果、セルフIDパケットの記憶に必要なRAMの使用容量を更に節約することに成功している。

【0086】

この場合、セルフIDパケットにエラーがあるか否かを各セルフIDパケットのパリティに基づき判断し、図14（B）に示すように、エラーがあるか否かを示すステータス情報であるHCEを、パケットのトレーラー（広義には制御情報）に付加することが望ましい。このようにすることで、セルフIDパケットのパリティを削除しても、セルフIDパケットにエラーがあったか否かを上層に適切に伝えることが可能になる。

【0087】

なお、バスに接続される複数のノードからのセルフIDパケットの中に1つでもエラーのセルフIDパケットがあると、たとえ他のセルフIDパケットにエラーが無くても、再度自己識別をやりなおす必要がある。従って、全てのセルフIDパケットに対して、エラーステータス情報としてのHCEは1つで十分となる。

【0088】

また、HCEは、図14（B）のようにトレーラー（フッター）に含ませることが望ましいが、HCEをヘッダに含ませたり、HCEをレジスタなどの所与の記憶手段に格納するようにしてもよい。

【0089】

また本実施形態では、図15（A）に示すように、RAMをヘッダ領域（広義には制御情報領域）とデータ領域に分離し、整形後のパケットのヘッダをヘッダ領域にデータをデータ領域に格納している。そして、データ領域に書き込まれるデータのアドレス（例えば先頭アドレス）を指すデータポインタをヘッダに付加している。

【0090】

このようにすることで、ヘッダとデータとがRAM上において混在しなくなるため、ヘッダやデータの取り扱いが容易になり、CPUの処理負荷を軽減できるようになる。

【0091】

なお、ヘッダ又はトレーラーに、データポインタの他に、データのサイズ（セルフIDパケット1～Nの全体のサイズ）を示すためのデータレングス情報を付加してもよい。

【0092】

また本実施形態では、図15（B）に示すように、セルフID期間に受信したパケットか否かを示すステータス情報であるBRを、パケットのトレーラーに付加している。

【0093】

即ち、図4（A）～（D）に示すように、物理層が扱うPHYパケットには、セルフIDパケットの他にリンクオンパケットとPHY構成パケットがある。そして本実施形態では、セルフIDパケットと他のPHYパケット（リンクオンパケット、PHY構成パケット）とをファームウェアなどが容易に区別できることが望ましい。図15（A）に示すように、セルフIDパケットにはデータポインタなどが付加されており、セルフIDパケットと他のPHYパケットとではファームウェアの取り扱いが異なったものになるからである。

【0094】

ところが、パケットの種類を区別するための情報としてIEEE1394において規格化されているt c o d eを用いても、処理対象となるパケットが、PHYパケットの中のどのパケットなのかを区別できない。

【0095】

本実施形態によれば、図15（B）に示すようにパケットのトレーラーに、セルフID期間に受信したパケットか否かを示すBRが付加される。このため、セルフIDパケットと他のPHYパケットとをファームウェア等が容易に区別できるようになり、処理負荷を軽減できる。

【0096】

なお、本実施形態では、図16に示すように、データ領域の中にセルフIDパケット専用領域を設け、この領域に整形後のパケットのデータ部分を書き込むようにしてもよい。このようにすれば、トポロジーマップの作成などの際の処理を簡易化でき、ファームウェア等の処理負荷を軽減できるようになる。

【0097】

3. 2 構成

さて、本実施形態におけるパケット整形の機能は図6のリンクコア20により実現され、パケット分離の機能はDMAC44により実現される。

【0098】

図17に、リンクコア20、FIFO34、DMAC44の構成の一例を示す。なお図17では、パケット整形やパケット分離に関係ない回路ブロックについては省略している。

【0099】

リンクコア20は、バス監視回路130、直列・並列変換回路132、パケット整形回路160を含む。そして、パケット整形回路160は、パケット診断回路142、シーケンサ167、バッファ168、セクタ170を含み、パケット診断回路142は、TAG生成回路162、ヘッダ&トレイラー生成回路164、エラーチェック回路166を含む。

【0100】

ここで、バス監視回路130は、PHYインターフェース10を介してPHYチップに接続される8ビット幅のデータバスD、2ビット幅のコントロールバスCTLを監視する回路である。

【0101】

直列・並列変換回路132は、データバスDのデータを32ビットのデータに変換する回路である。例えば、転送速度が400Mbpsの場合には8ビットのデータが32ビットのデータに、200Mbpsの場合には4ビットのデータが32ビットのデータに、100Mbpsの場合には2ビットのデータが32ビットのデータに変換される。

【0102】

パケット診断回路142は、セルフIDパケットなどのパケットを診断する回路である。TAG生成回路162は、ヘッダ、データ、トレーラーなどを区別するためのTAGを生成する回路であり、ヘッダ&トレーラー生成回路164は、図13(B)～図15(B)などで説明したヘッダ及びトレーラー(フッター)を生成する回路である。また、エラーチェック回路166は、パケットに含まれるパリティなどのエラーチェック情報をチェックしてエラーを検出する回路である。

【0103】

シーケンサ167は各種の制御信号を生成するものである。バッファ168、セクタ170は、直列・並列変換回路132からのDI、パケット診断回路142からのヘッダ及びトレーラー、DMAC44からのデータポインタのいずれかを、パケット診断回路142からの信号SELにより選択するためのものである。

【0104】

FIFO34は、リンコア20からの出力データであるRDの位相と、RAM80への書き込みデータであるWDATAの位相とを調整するためのバッファとして機能するものであり、FIFO状態判断回路35を含む。FIFO状態判断回路35は、FIFOが空になると、EMPTYをアクティブにし、FIFOがフルになると、FULLをアクティブにする。

【0105】

DMAC44は、パケット分離回路180、アクセス要求実行回路190、アクセス要求発生回路192を含む。

【0106】

パケット分離回路180は、パケット整形回路160により整形されたパケットを分離して、ヘッダ及びトレーラーをRAM80のヘッダ領域に、データをデータ領域に書き込むための回路である(図7参照)。パケット分離回路180は、TAG判別回路182、ポインタ更新回路184、アドレス発生回路188を含む。

【0107】

TAG判別回路182は、TAG生成回路162により生成されたTAG (DTAG) を判別する回路である。

【0108】

ポインタ更新回路184は、TAG判別回路182の出力を受け、RAM80にヘッダやデータを書き込むためのヘッダポインタやデータポインタを更新するための回路である。

【0109】

アドレス発生回路188は、ポインタ更新回路184の出力を受け、RAM80への書き込みアドレスWADRを発生する回路である。

【0110】

アクセス要求実行回路190は、リンクコア20からのアクセス要求を実行するための回路である。アクセス要求実行回路190は、FIFO状態判断回路35からのFULLがアクティブになると、FFULLをアクティブにする。パケット整形回路160内のシーケンサ167は、FFULLがアクティブでないことを条件に、RD (RxData) のストロブ信号であるRDSをアクティブにする。

【0111】

なおRFAILは、受信における失敗を、シーケンサ167がアクセス要求実行回路190に対して知らせるための信号である。

【0112】

アクセス要求発生回路192は、RAM80へのアクセス要求を発生するための回路である。アクセス要求発生回路192は、バッファマネージャ70からの書き込みアクノリッジメントであるWACKやFIFO状態判断回路35からのEMPTYを受け、書き込み要求であるWREQをバッファマネージャ70に出力する。

【0113】

3. 3 動作

次に、本実施形態の動作の詳細について図18のタイミング波形図などを用い

て説明する。

【0114】

まず、リンクコア20の動作について説明する。

【0115】

バス監視回路130は、セルフID期間（バスリセットから1回目のサブアクションギャップまでの期間）中か否かを判断し、セルフID期間中の場合には図18のC1に示すように、バスリセット中であることを示す信号BRIPをHレベルにする。

【0116】

セルフID期間中か否かは以下のようにして検出する。まず、バス監視回路130が、PHYチップからの2ビットのCTLを監視する。そして図19（A）に示すようにCTLが（01）の場合には、図19（B）に示すようにPHYチップからデータバスDを介してステータス情報が送られてくると判断する。このPHYチップからDを介して送られてくるステータス情報の中のステータスビットの内容を調べることで、図19（C）に示すようにバスリセットがなされたか否か（ビット2）、サブアクションギャップが来たか否か（ビット1）などを判断できる。そして、バス監視回路130は、バスリセットがなされたと判断すると、図18に示すようにBRIPをHレベルにし、サブアクションギャップが来たと判断すると、BRIPをLレベルに戻す。

【0117】

このBRIPのレベルを監視することで、パケット診断回路142は、セルフID期間中か否かを知ることができるようになる。そして、このセルフID期間に送られてきたパケットをセルフIDパケットと見なし、前述のパケット整形を行う。

【0118】

セルフID期間に入ると、まず、パケット診断回路142のヘッダ&トレイラー生成回路164がヘッダを生成する。このヘッダは、バッファ168を介してセレクタ170に入力され、パケット診断回路142からの信号SELに基づきセレクタ170がこのヘッダを選択する。これにより、図18のC2に示すよう

に、RDとしてヘッダがFIFO34に出力されることになる。

【0119】

なお図20(A)に、パケットがセルフIDパケットである場合にヘッダ&トレイラー生成回路164が生成するヘッダ及びトレイラーのフォーマットを示す。同図において網掛けとなっている部分がトレイラーであり、それ以外がヘッダである。

【0120】

また図20(B)に、パケットがセルフIDパケット以外のPHYパケットである場合に、ヘッダ&トレイラー生成回路164が生成するヘッダ及びトレイラーのフォーマットも示す。

【0121】

ヘッダがRDとして出力されると、次に、DMAC44からのデータポインタがバッファ168を介してセレクタ170に入力され、セレクタ170がこれを選択する。これにより、図18のC3に示すように、RDとしてデータポインタがFIFO34に出力される。

【0122】

さて、図19(B)に示すように、CTLが(10)である場合には受信状態になり、PHYチップからデータバスDを介してセルフIDパケットが送られてくる。直列・並列変換回路132は、これらのセルフIDパケットのデータを32ビットのデータであるDIに変換し、パケット診断回路142及びバッファ168に出力する。

【0123】

なお、DIEは、DIのデータが有効か無効かを示す信号である(DIEがHレベルの時に有効)。このDIEを調べることでパケット診断回路142はパケットの区切りを知ることができる。また、DISは、DIの取り込みタイミングを知らせるためのストローク信号である。

【0124】

データポインタがRDとして出力されると、次に、直列・並列変換回路132からの上記DI(一連のセルフIDパケット)がバッファ168を介してセレク

タ170に入力され、セクタ170がこれを選択する。これにより、C4に示すように、RDとして一連のセルフIDパッケージがFIFO34に出力される。

【0125】

なお、この際に本実施形態では、図14(A)で説明したように、セルフIDパッケージの第2クワドレットについては削除し、FIFO34に出力しないようにしている。また、これらの第2クワドレットについてはパッケージ診断回路142のエラーチェック回路166がチェックする。そして、図14(B)で説明したように、一連のセルフIDパッケージの中に1つでもエラーのパッケージがあった場合には、エラーチェック回路166は、トレーラーの中にエラーステータス情報HCEを付加するようにヘッダ&トレーラー生成回路164に指示する。

【0126】

DIがRDとして出力されると、次に、ヘッダ&トレーラー生成回路164からのトレーラーがバッファ168を介してセクタ170に入力され、セクタ170がこれを選択する。これにより、C5に示すように、RDとしてトレーラーがFIFO34に出力される。

【0127】

このトレーラーは、図20(A)に示すように、データのサイズを表すData length、セルフID期間中のパッケージか否かを示すBR、エラーステータス情報であるHCEを含む。

【0128】

なお、処理対象となるパッケージがセルフID期間中のセルフIDパッケージである場合には、図20(A)に示すようにBRは1となり、セルフID期間外のリンクオンパッケージやPHY構成パッケージである場合には、図20(B)に示すようにBRは0になる。このようにすることで、tcodeが同じ0xEであっても、ファームウェアはこれらのパッケージを区別できるようになる。

【0129】

また、セルフIDパッケージの中に1つでもエラーのパッケージがあった場合には、HCEが1になる。これにより、ファームウェア等は、セルフIDパッケージにエラーがあったか否かを簡易に検出でき、自己識別を再度やり直すなどの処理が

可能になる。

【0130】

さて、TAG生成回路162は、RDとして出力される情報を区別するためのTAGを生成している。本実施形態では図21に示すようにTAGは2ビットであり、(00)、(01)、(10)、(11)は、各々、ヘッダ、トレイラー、データ、スタート(ヘッダの最初)を表す。従って、例えば図18では、(11)、(00)、(10)、(10)、…、(01)というようにTAGが変化する。FIFO34には、この2ビットのTAGと32ビットのRDとからなる34ビットのデータが入力されることになる。

【0131】

なお、図22に、セルフID期間外において、セルフIDパケット以外のPHYパケット(リンクオンパケット、PHY構成パケット)を整形する際のタイミング波形図を示す。パケット診断回路142は、BRIPがLレベルである場合には、セルフID期間外であると判断する。そして、この場合には、信号SELを制御して、ヘッダの生成後にデータポインタを選択しないようにする。即ち図18のC3とは異なり、図22のD1に示すようにヘッダにデータポインタを付加する処理は行われず。また図18のC4と異なり、図22のD2に示すように複数のPHYパケットを1つにパッケージングする処理も行われず。また、図20(B)に示すように、セルフID期間中か否かを表すBRが0に設定される。またTAGは、(11)、(10)、(01)と変化する。

【0132】

次に、DMAC44の動作について説明する。

【0133】

パケット分離回路180に含まれるTAG判別回路182は、RAM80への書き込みデータであるWDATAと共にFIFO34から出力されるDTAGを判別し、WDATAが、スタート(ヘッダの最初)、ヘッダ、データ、トレイラーのいずれなのかを判定する。そして、ポインタ更新回路184は、この判定結果に基づいて、ヘッダポインタやデータポインタの更新を行う。次に、アドレス発生回路188は、更新されたヘッダポインタやデータポインタに基づいて、W

DATAの書き込みアドレスであるWADRを発生する。

【0134】

より具体的には、例えば、WDATAがスタート又はヘッダであるとDTAGに基づき判定された場合は、ポインタ更新回路184が、ヘッダポインタのインクリメント（広義には更新）を行う。アドレス発生回路188は、インクリメントされるヘッダポインタに基づきRAM80のアドレスであるWADRを発生する。

【0135】

次に、WDATAがデータであるとDTAGに基づき判定された場合は、ポインタ更新回路184が、データポインタのインクリメントを行う。アドレス発生回路188は、インクリメントされるデータポインタに基づきWADRを発生する。最後に、WDATAがトレーラーであるとDTAGに基づき判定された場合は、ポインタ更新回路184が、今度は、ヘッダポインタのインクリメントを行う。

【0136】

以上のようにして、パケットを分離してヘッダ領域とデータ領域に書き込むことが可能になる。

【0137】

特に本実施形態では、ヘッダに付加されるデータポインタが、ポインタ更新回路184からパケット整形回路160に伝えられる。そしてパケット整形回路160が、この伝えられたデータポインタをパケットのヘッダに付加する。このようにすることで、ヘッダ領域からヘッダを読み出したファームウェアなどが、そのヘッダに対応するデータのデータ領域での格納アドレスを容易に知ることができるようになる。また、データポインタの付加は、パケット整形回路160により行われ、DMAC44はこれに関与する必要がない。従って、DMAC44がRAM80へのデータ書き込み処理に専念できるようになり、DMAC44の回路構成や処理を簡素化できるようになる。

【0138】

なお、RAM80を分離する領域の境界、例えばヘッダ領域とデータ領域の境

界の設定は、CPUインターフェース60を介してCPU66（ファームウェア等）が、図6のレジスタ46に含まれるポインタ設定レジスタに対して、境界のアドレスを指すポインタを設定することで実現される。

【0139】

また、データ領域が複数の領域に分離される場合（アイソクロナス転送用と非同期転送用の領域に分離される場合、第1、第2の非同期転送用の領域に分離される場合等）には、各々が各領域を指す複数のデータポインタを用意することが望ましい。より具体的には、図17に示すように、DMAC44が、複数のデータポインタ、例えば第1、第2のデータポインタをパケット整形回路160に対して渡すようにする（3個以上のデータポインタを渡してもよい）。このようにすることで、デジタルカメラにおける動画像データをアイソクロナス転送用データ領域に連続的に格納したり、プリンタにおける印字データを第2の非同期転送用データ領域に連続的に格納したりすること（第1の非同期転送用データ領域にはコマンドデータ、ステータスデータなどの制御用データを格納する）が可能になる。

【0140】

アクセス要求発生回路192は、FIFO状態判断回路35からのEMPTYや、バッファマネージャ70からのWACKに基づいて、WREQを生成し、バッファマネージャ70に出力する。バッファマネージャ70の調停回路74は、このWREQや、PF用のDMAC54や、CPUインターフェース60からWREQに基づいてバスの調停を行うことになる。

【0141】

4. 電子機器

次に、本実施形態のデータ転送制御装置を含む電子機器の例について説明する。

【0142】

例えば図23（A）に電子機器の1つであるプリンタの内部ブロック図を示し、図24（A）にその外観図を示す。CPU（マイクロコンピュータ）510はシステム全体の制御などを行う。操作部511はプリンタをユーザが操作するた

めのものである。ROM516には、制御プログラム、フォントなどが格納され、RAM518はCPU510のワーク領域として機能する。表示パネル519はプリンタの動作状態をユーザに知らせるためのものである。

【0143】

PHYチップ502、データ転送制御装置500を介して、パーソナルコンピュータなどの他のノードから送られてきた印字データは、バス504を介して印字処理部512に直接送られる。そして、印字データは、印字処理部512にて所与の処理が施され、プリントヘッダなどからなる印字部（データを出力するための装置）514により紙に印字されて出力される。

【0144】

図23（B）に電子機器の1つであるスキャナの内部ブロック図を示し、図24（B）にその外観図を示す。CPU520はシステム全体の制御などを行う。操作部521はスキャナをユーザが操作するためのものである。ROM526には制御プログラムなどが格納され、RAM528はCPU520のワーク領域として機能する。

【0145】

光源、光電変換器などからなる画像読み取り部（データを取り込むための装置）522により原稿の画像が読み取られ、読み取られた画像のデータは画像処理部524により処理される。そして、処理後の画像データがバス505を介してデータ転送制御装置500に直接送られる。データ転送制御装置500は、この画像データにヘッダなどを付加することでパケットを生成し、PHYチップ502を介してパーソナルコンピュータなどの他のノードに送信する。

【0146】

図23（C）に電子機器の1つであるCD-Rドライブの内部ブロック図を示し、図24（C）にその外観図を示す。CPU530はシステム全体の制御などを行う。操作部531はCD-Rをユーザが操作するためのものである。ROM536には制御プログラムなどが格納され、RAM538はCPU530のワーク領域として機能する。

【0147】

レーザ、モータ、光学系などからなる読み取り&書き込み部（データを取り込むための装置又はデータを記憶するための装置）533によりCD-R532から読み取られたデータは、信号処理部534に入力され、エラー訂正処理などの所与の信号処理が施される。そして、信号処理が施されたデータが、バス506を介してデータ転送制御装置500に直接送られる。データ転送制御装置500は、このデータにヘッダなどを付加することでパケットを生成し、PHYチップ502を介してパーソナルコンピュータなどの他のノードに送信する。

【0148】

一方、PHYチップ502、データ転送制御装置500を介して、他のノードから送られてきたデータは、バス506を介して信号処理部534に直接送られる。そして、信号処理部534によりこのデータに所与の信号処理が施され、読み取り&書き込み部533によりCD-R532に記憶される。

【0149】

なお、図23（A）、（B）、（C）において、CPU510、520、530の他に、データ転送制御装置500でのデータ転送制御のためのCPUを別に設けるようにしてもよい。

【0150】

本実施形態のデータ転送制御装置を電子機器に用いることで、高速なデータ転送が可能になる。従って、ユーザがパーソナルコンピュータなどによりプリントアウトの指示を行った場合に、少ないタイムラグで印字が完了するようになる。また、スキャナへの画像取り込みの指示の後に、少ないタイムラグで読み取り画像をユーザは見るようになる。また、CD-Rからのデータの読み取りや、CD-Rへのデータの書き込みを高速に行うことができるようになる。更に、例えば1つのホストシステムに複数の電子機器を接続して利用したり、複数のホストシステムに複数の電子機器を接続して利用したりすることも容易になる。

【0151】

また本実施形態のデータ転送制御装置を電子機器に用いることで、CPU上で動作するファームウェアの処理負荷が軽減され、安価なCPUや低速のバスを用

いることが可能になる。更に、パケットを記憶する RAM を小容量化できる。従って、電子機器の低コスト化、小規模化を図ることが可能になる。

【0152】

なお本実施形態のデータ転送制御装置を適用できる電子機器としては、上記以外にも例えば、種々の光ディスクドライブ（CDROM、DVD）、光磁気ディスクドライブ（MO）、ハードディスクドライブ、TV、VTR、ビデオカメラ、オーディオ機器、電話機、プロジェクタ、パーソナルコンピュータ、電子手帳、ワードプロセッサなど種々のものを考えることができる。

【0153】

なお、本発明は本実施形態に限定されず、本発明の要旨の範囲内で種々の変形実施が可能である。

【0154】

例えば、本発明のデータ転送制御装置の構成は、図 6 に示す構成が特に望ましいが、これに限定されるものではない。例えば、図 8 に示すような構成を採用することも可能である。

【0155】

また、本発明は、IEEE 1394 規格でのデータ転送に適用されることが特に望ましいが、これに限定されるものではない。例えば IEEE 1394 と同様の思想に基づく規格や IEEE 1394 を発展させた規格におけるデータ転送にも本発明は適用できる。

【0156】

【図面の簡単な説明】

【図 1】

図 1 (A)、(B)、(C) は、非同期転送とアイソクロナス転送について説明するための図である。

【図 2】

図 2 (A)、(B) は、ツリー識別について説明するための図である。

【図 3】

自己識別について説明するための図である。

【図4】

図4 (A)、(B)、(C)、(D)は、セルフIDパケットなどの物理層のパケットのフォーマットを示す図である。

【図5】

IEEE 1394のプロトコル構成について示す図である。

【図6】

本実施形態のデータ転送制御装置の構成例を示す図である。

【図7】

ヘッダ(制御情報)領域とデータ領域の分離について説明するための図である。

【図8】

本実施形態の比較例の構成例について示す図である。

【図9】

図8の構成によるデータ転送の手法について説明するための図である。

【図10】

データ転送の手法の他の例について説明するための図である。

【図11】

本実施形態のデータ転送の手法について説明するための図である。

【図12】

セルフID期間について説明するための図である。

【図13】

図13 (A)、(B)は、各々、比較例、本実施形態のパケット整形について説明するための図である。

【図14】

図14 (A)、(B)は、セルフIDパケットのパリティの部分削除する手法について説明するための図である。

【図15】

図15 (A)は、データポインタをヘッダに付加する手法を、図15 (B)は、セルフID期間のパケットか否かを示すBRをトレーラーに付加する手法を説

明するための図である。

【図 16】

データ領域にセルフ ID パケット専用の領域を設ける手法について説明するための図である。

【図 17】

リンクコア（パケット整形回路）と DMAC（RF 用）の構成例について示す図である。

【図 18】

セルフ ID 期間でのパケット整形の詳細な処理例について説明するためのタイミング波形図である。

【図 19】

図 19（A）、（B）、（C）は、コントロールバス CTL、データバス D、ステータス情報（ステータスビット）について説明するための図である。

【図 20】

図 20（A）、（B）は、PHY パケットのフォーマットを示す図である。

【図 21】

TAG について説明するための図である。

【図 22】

セルフ ID 期間外でのパケット整形の詳細な処理例について説明するためのタイミング波形図である。

【図 23】

図 23（A）、（B）、（C）は、種々の電子機器の内部ブロック図の例である。

【図 24】

図 24（A）、（B）、（C）は、種々の電子機器の外観図の例である。

【符号の説明】

- 10 PHY インターフェース
- 20 リンクコア
- 22 レジスタ

- 30 FIFO (ATF)
- 32 FIFO (ITF)
- 34 FIFO (RF)
- 35 FIFO状態判断回路
- 40 DMAC (ATF用)
- 42 DMAC (ITF用)
- 44 DMAC (RF用)
- 50 ポートインターフェース
- 52 FIFO (PF)
- 54 DMAC (PF用)
- 56 レジスタ
- 60 CPUインターフェース
- 62 アドレスデコーダ
- 63 データ同期化回路
- 64 割り込みコントローラ
- 66 CPU
- 68 クロック制御回路
- 70 バッファマネージャ
- 72 レジスタ
- 74 調停回路
- 76 シーケンサ
- 80 RAM (パケット記憶手段)
- 90、92、94 バス (第1のバス)
- 96、98 バス (第2のバス)
- 100、102、104、105、
- 106、107、108、109 バス (第3のバス)
- 110 バス (第4のバス)
- 120 データ転送制御装置
- 122 PHYチップ

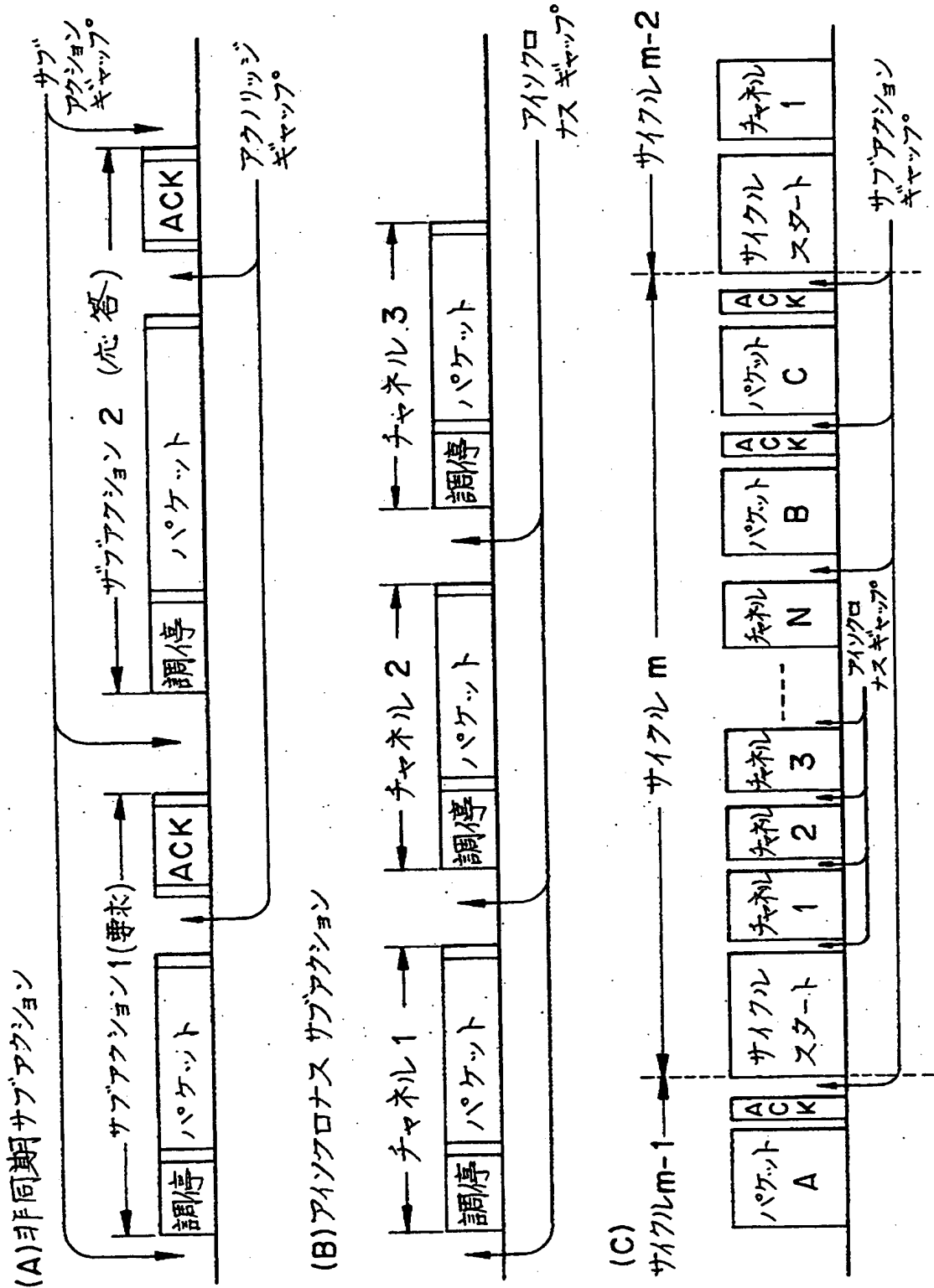
- 124 アプリケーション層のデバイス
- 130 バス監視回路
- 132 直列・並列変換回路
- 142 パケット診断回路
- 160 パケット整形回路
- 162 TAG生成回路
- 164 ヘッダ&トレイラー生成回路
- 166 エラーチェック回路
- 167 シーケンサ
- 168 バッファ
- 170 セレクタ
- 180 パケット分離回路
- 182 TAG判別回路
- 184 ポインタ更新回路
- 188 アドレス発生回路
- 190 アクセス要求実行回路
- 192 アドレス要求発生回路

特平 10-321540

【書類名】

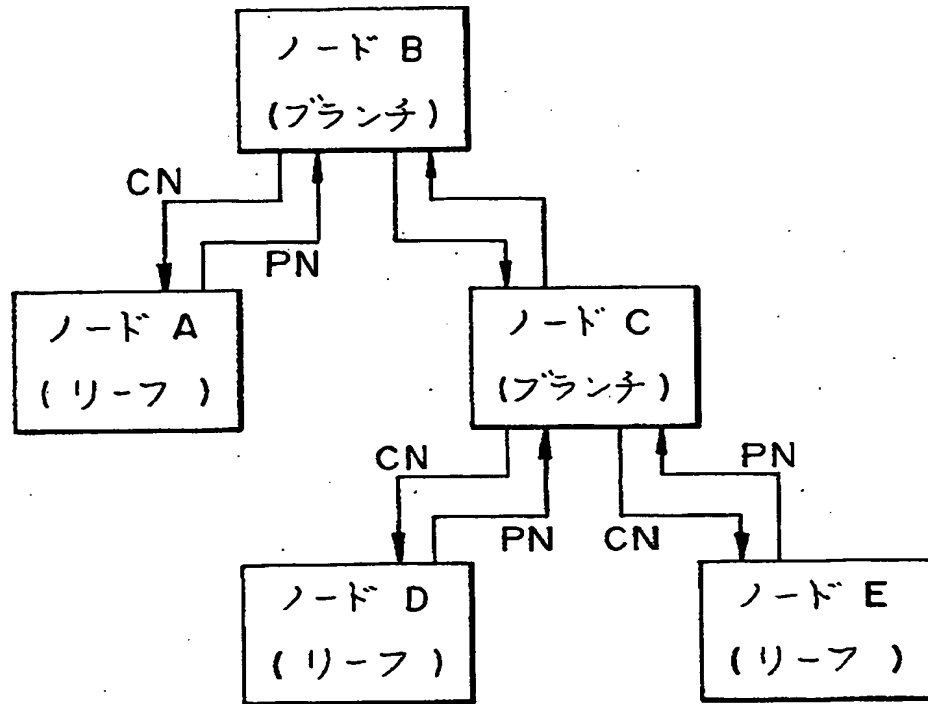
図面

【図 1】

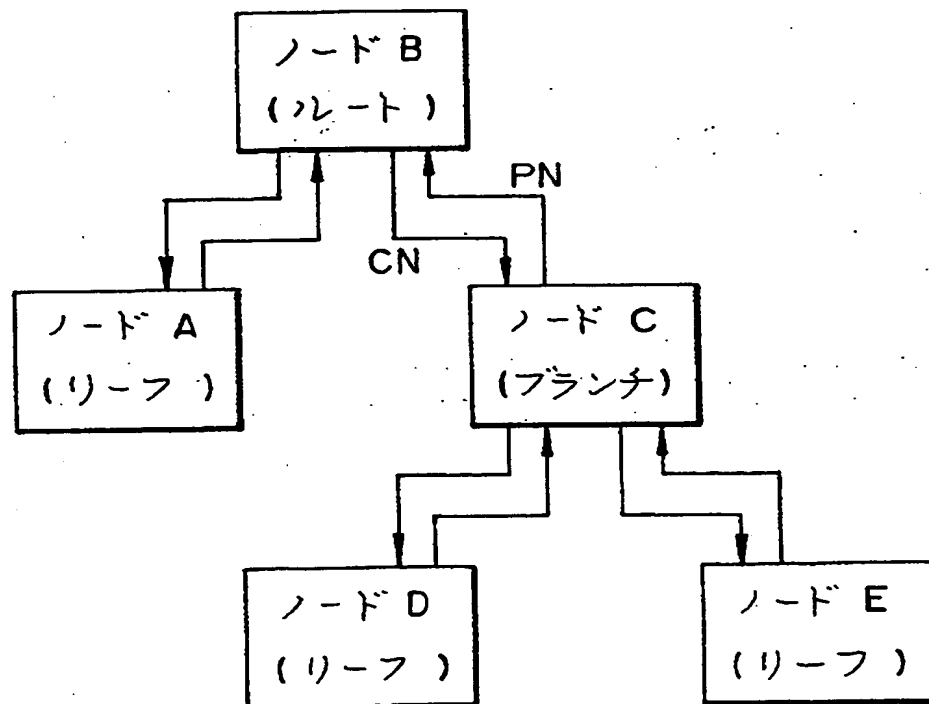


【図 2】

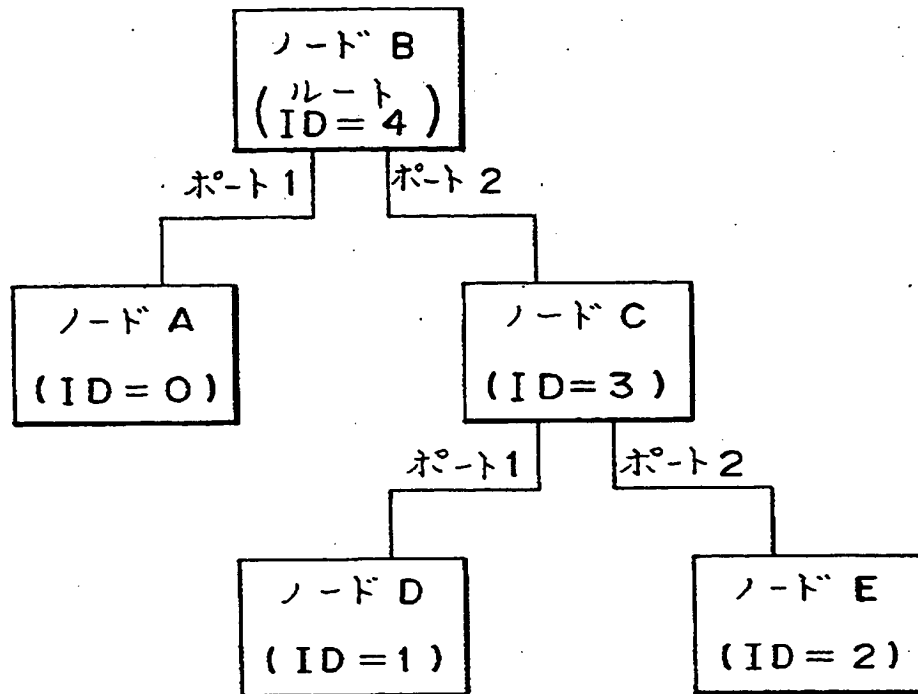
(A)



(B)



【図3】



【図 4】

セルフ ID パケット # 0

		b.31													b.0	
(A)	0	10	PHY_ID	0	L	gap_cnt	sp	del	C	pwr	p0	p1	p2	i	m	
	1	最初の32ビットの反転														

セルフ ID パケット # 1, # 2, # 3

		b.31														b.0	
(B)	0	10	PHY_ID	1	n	rsv	pa	pb	pc	pd	pe	pf	pg	ph	r	m	
	1	最初の32ビットの反転															

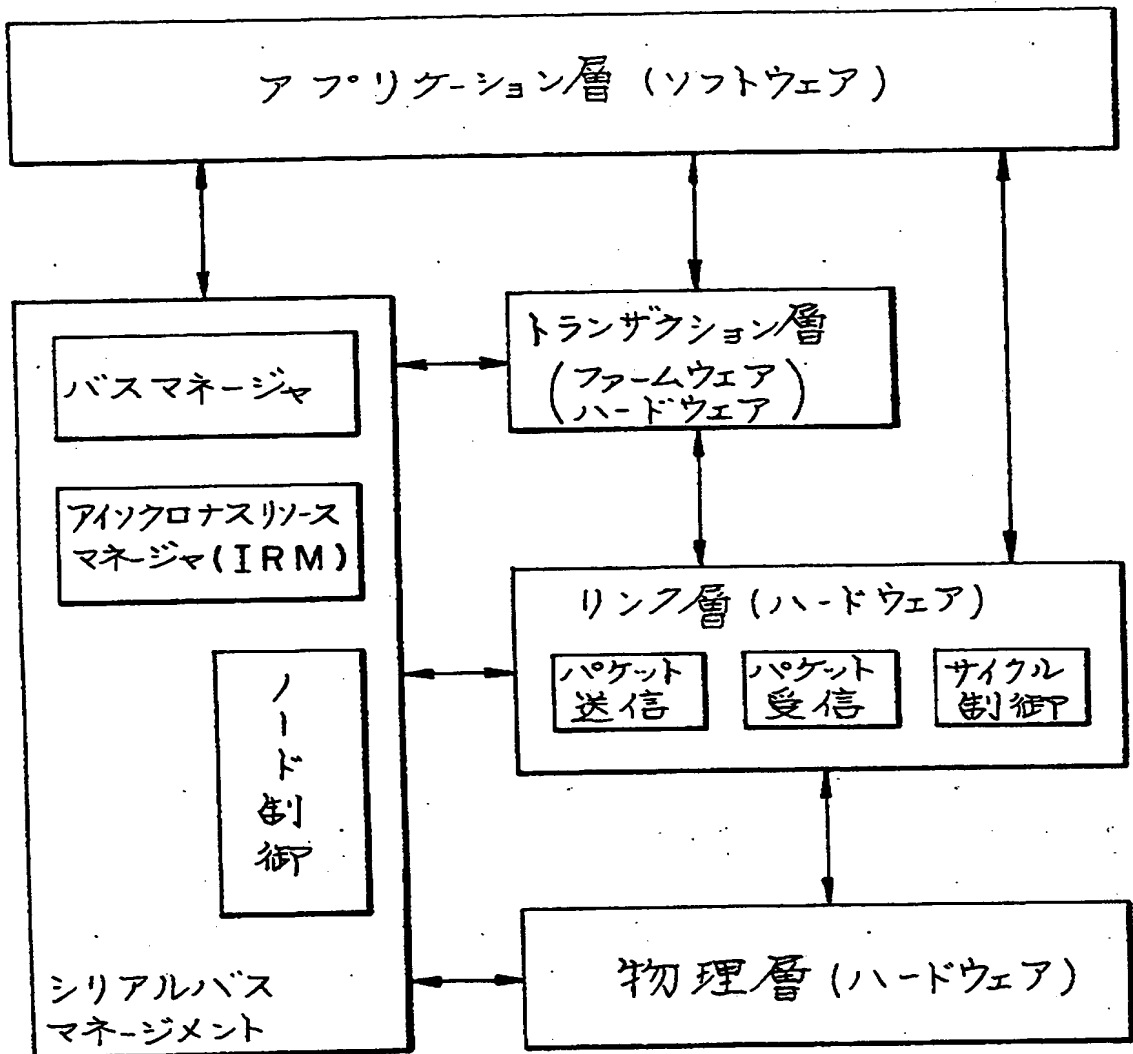
リンクオンパケット

	b.31							b.0	
(C)	0	01	PHY_ID	0000	0000	0000	0000	0000	0000
	1	最初の32ビットの反転							

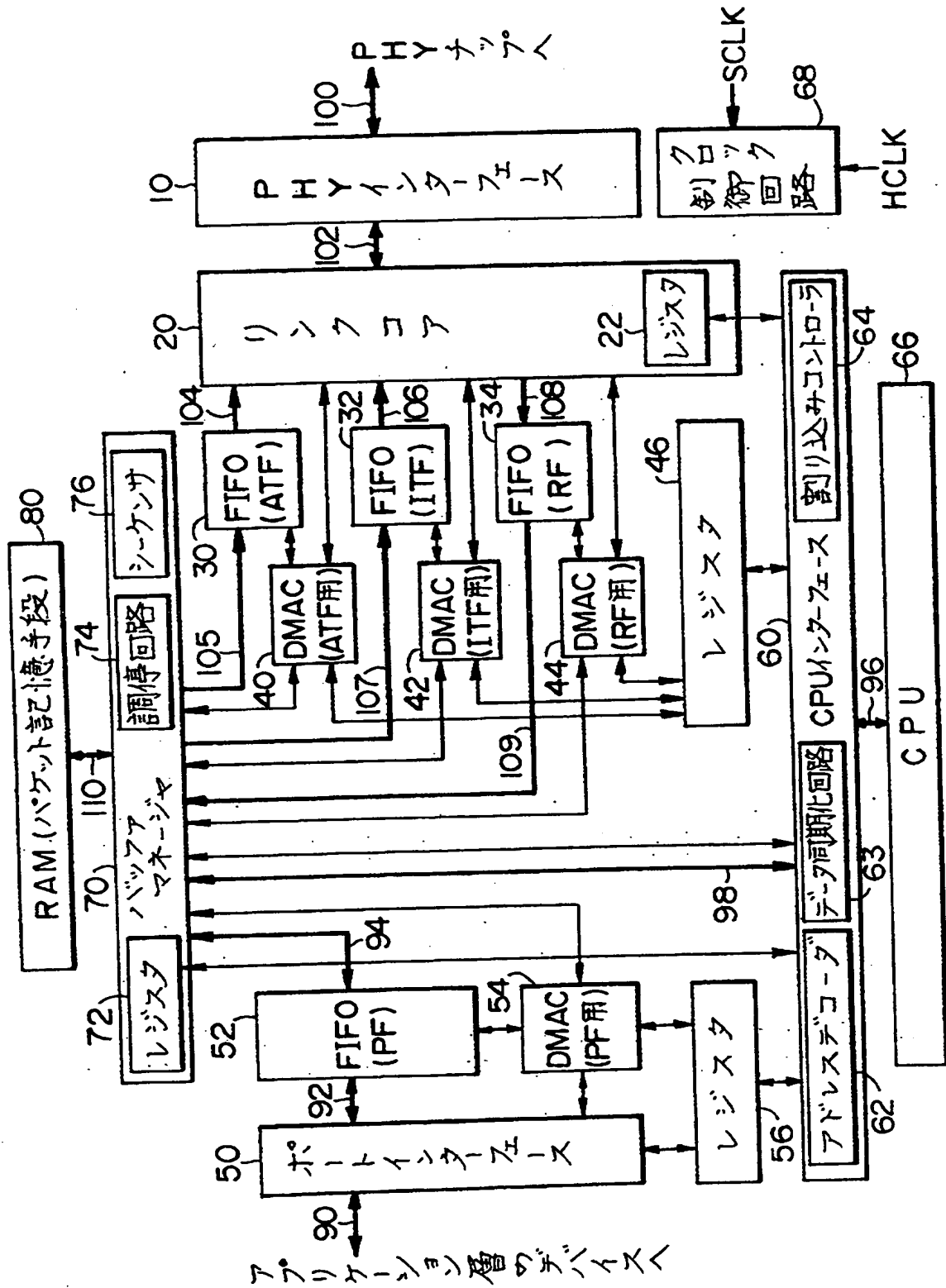
PHY 構成パケット

	b.31								b.0	
(D)	0	00	PHY_ID	R	T	gap_cnt	0000	0000	0000	0000
	1	最初の32ビットの反転								

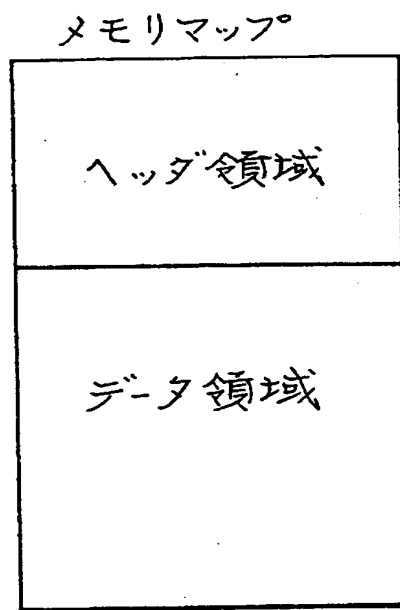
【図5】



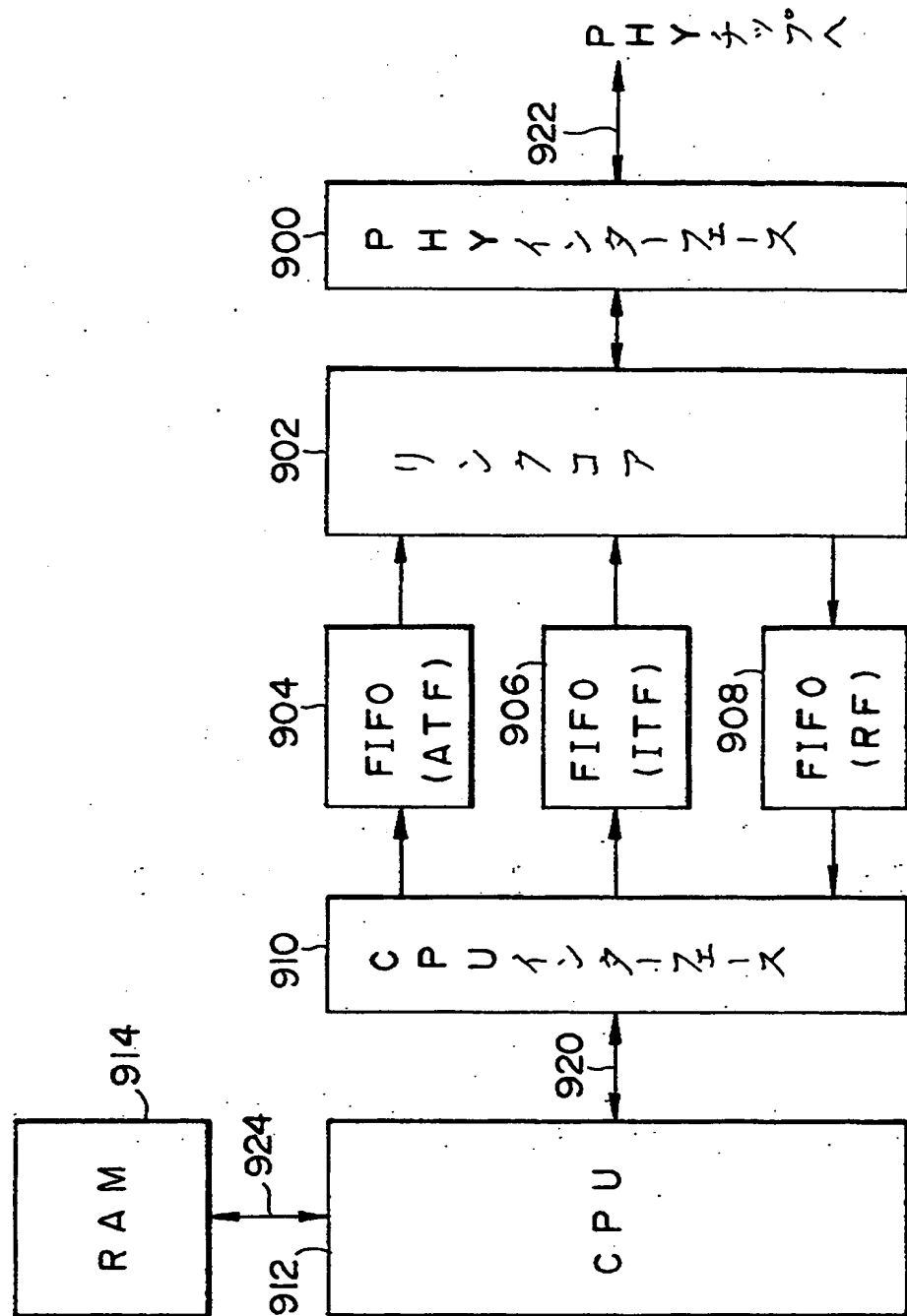
【図 6】



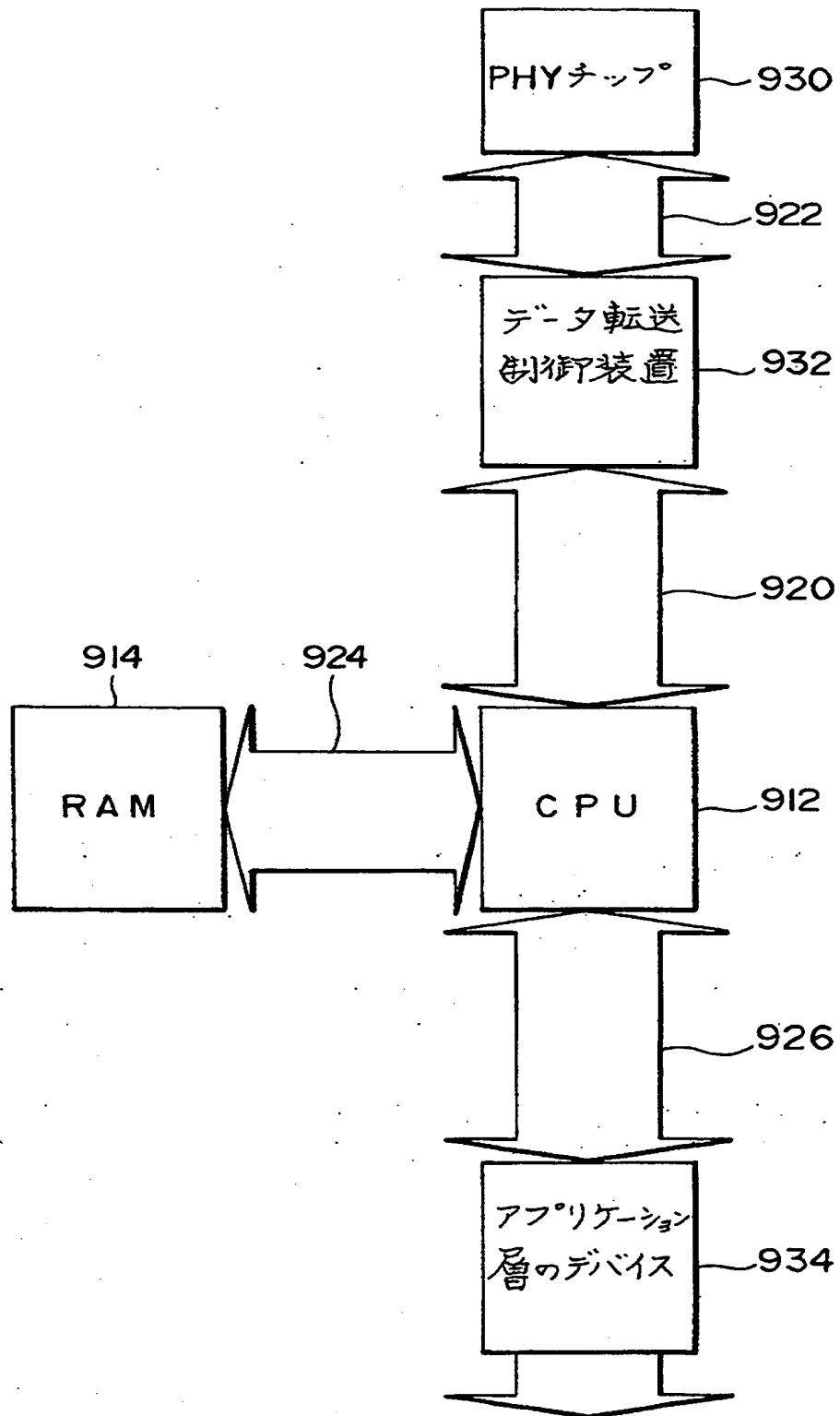
【図 7】



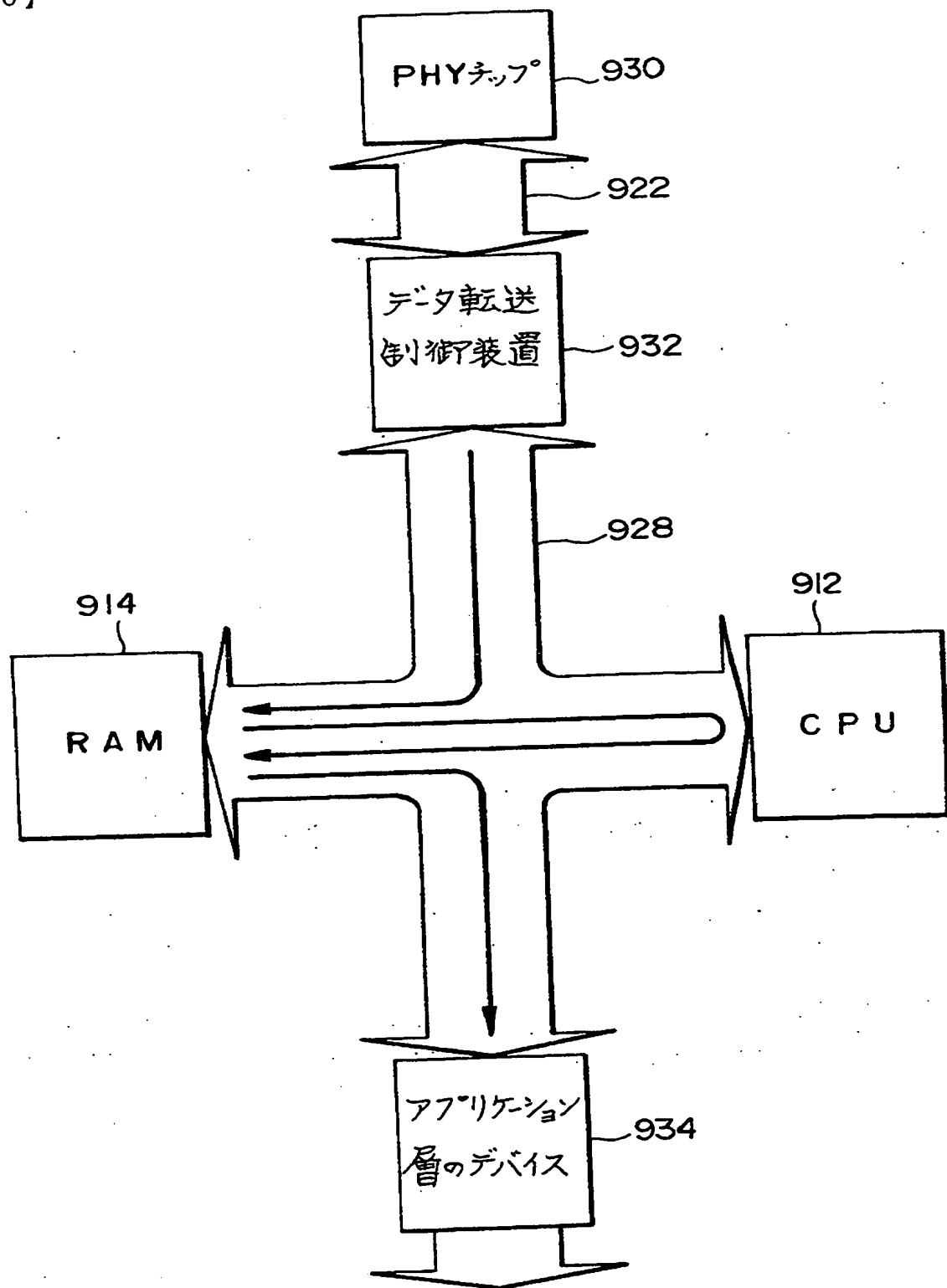
【図 8】



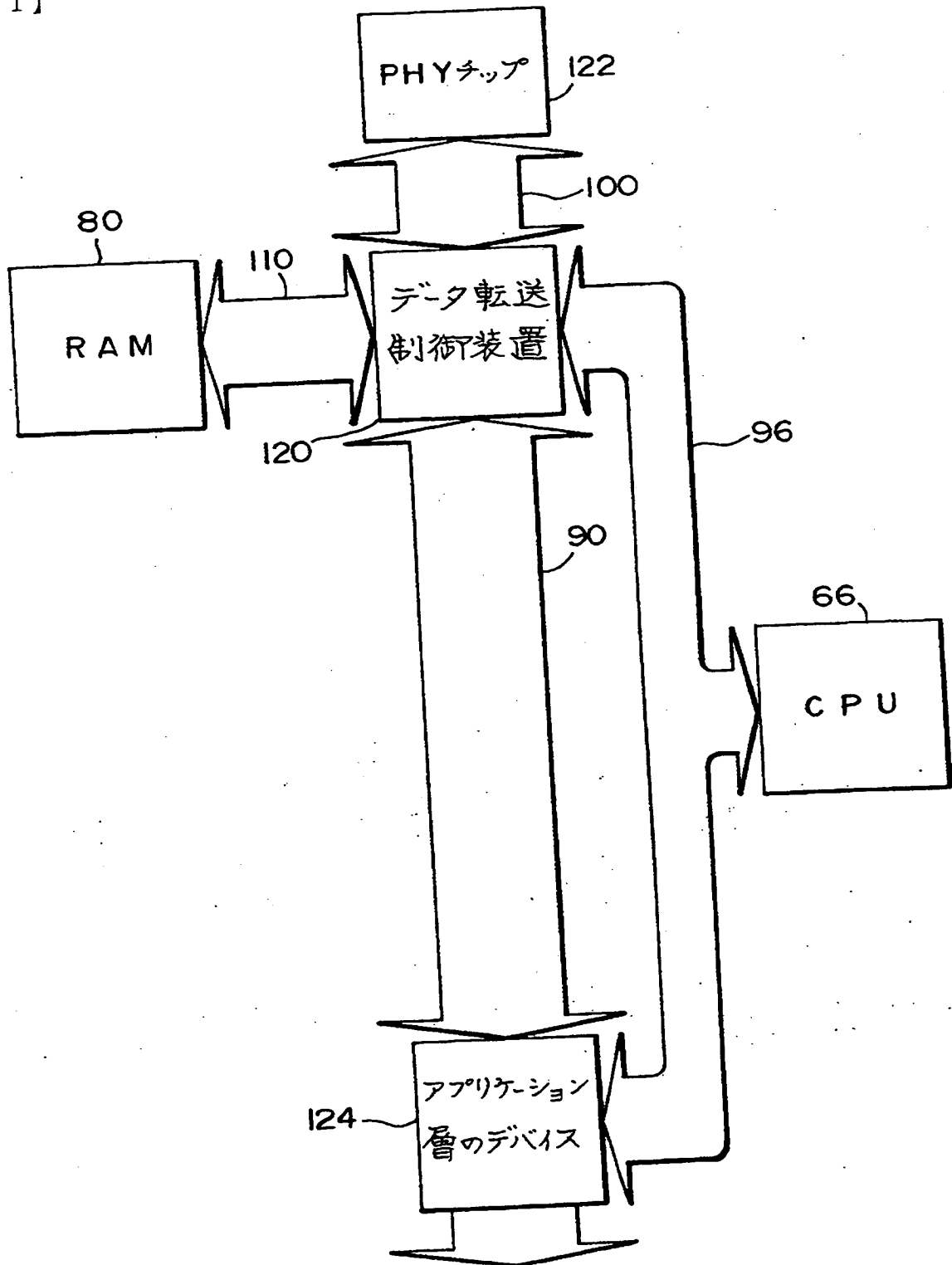
【図9】



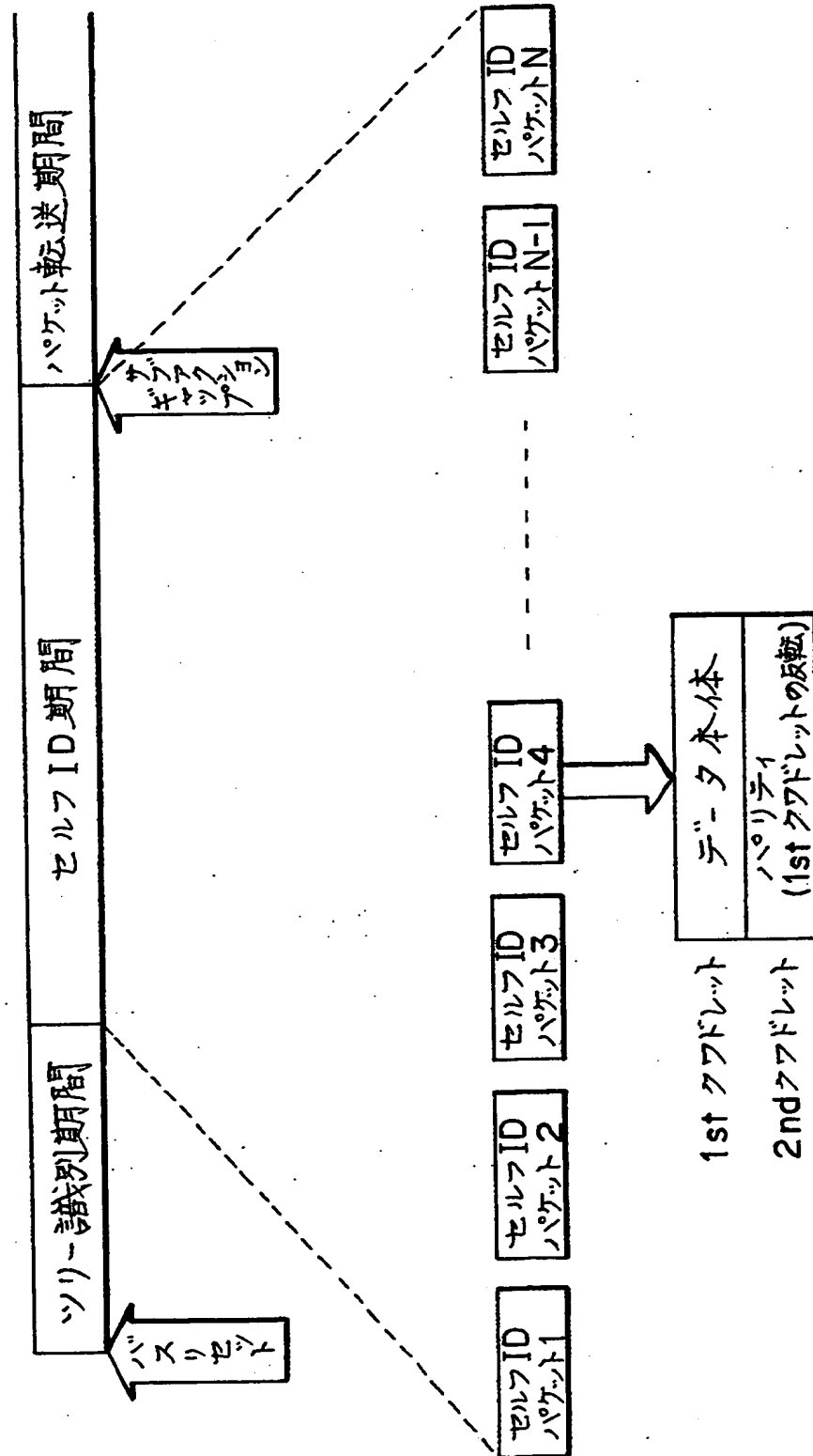
【図 10】



【図 11】

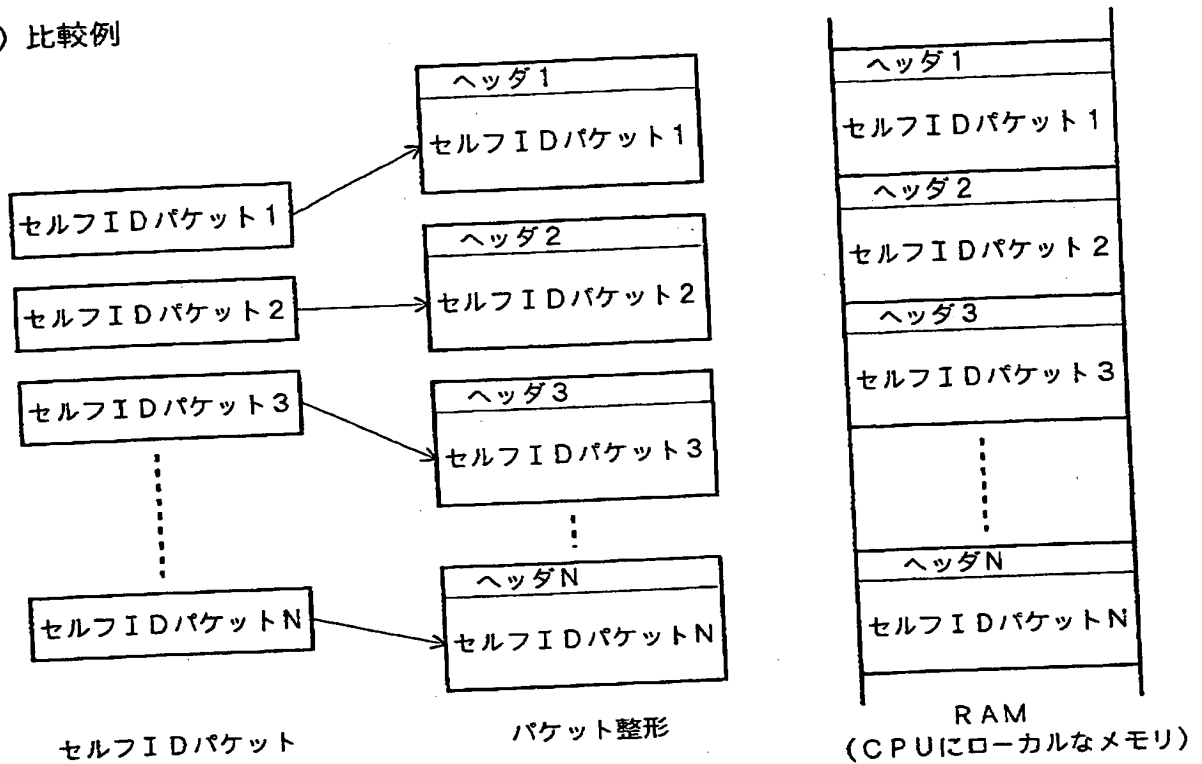


【図 12】

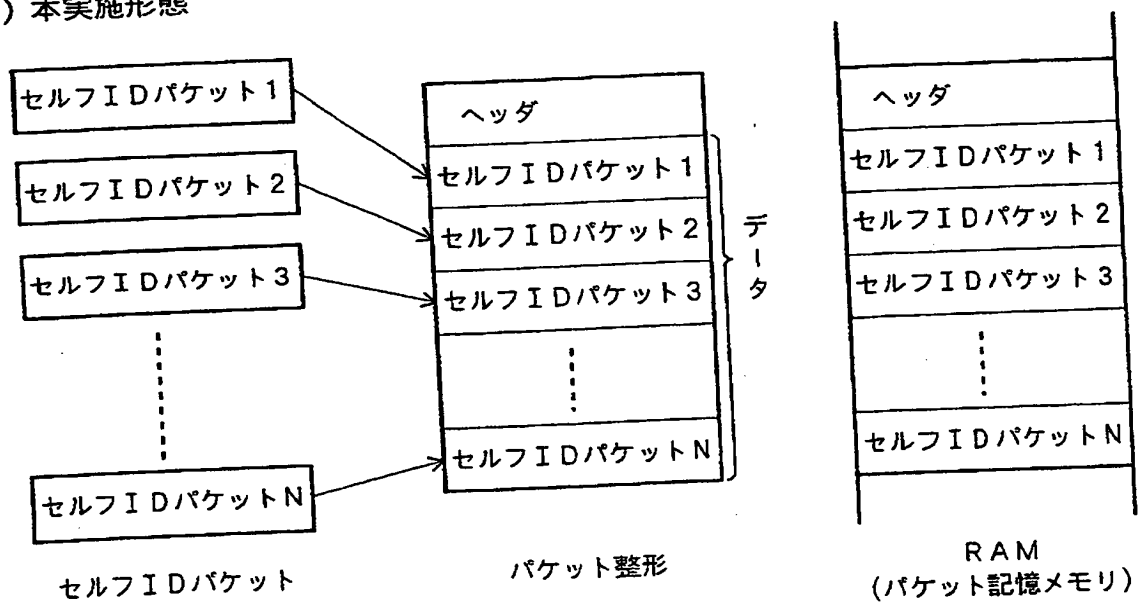


【図 13】

(A) 比較例

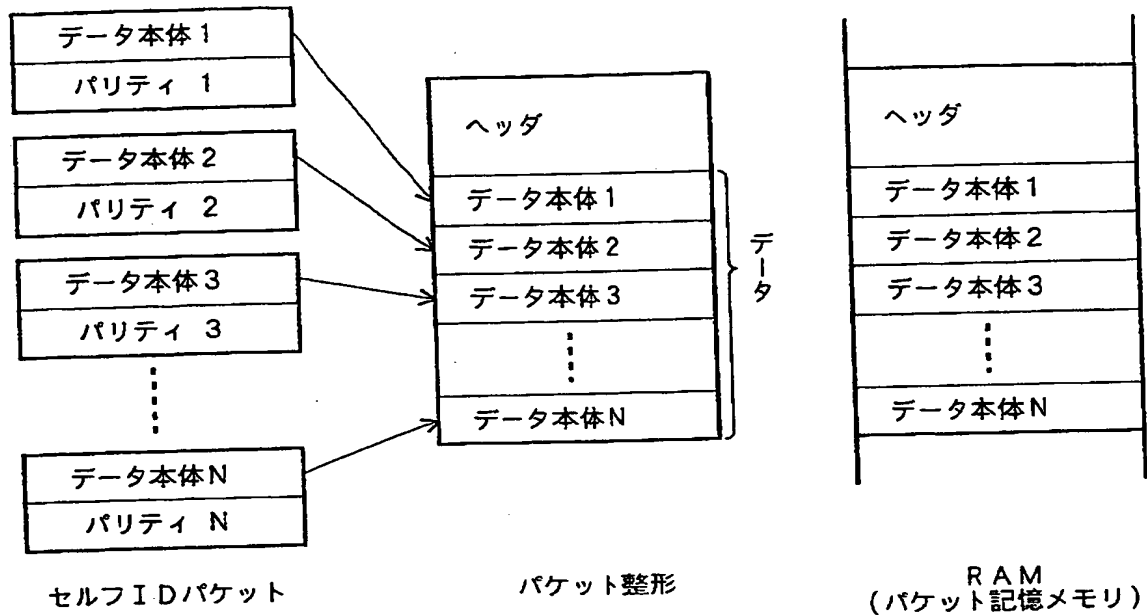


(B) 本実施形態

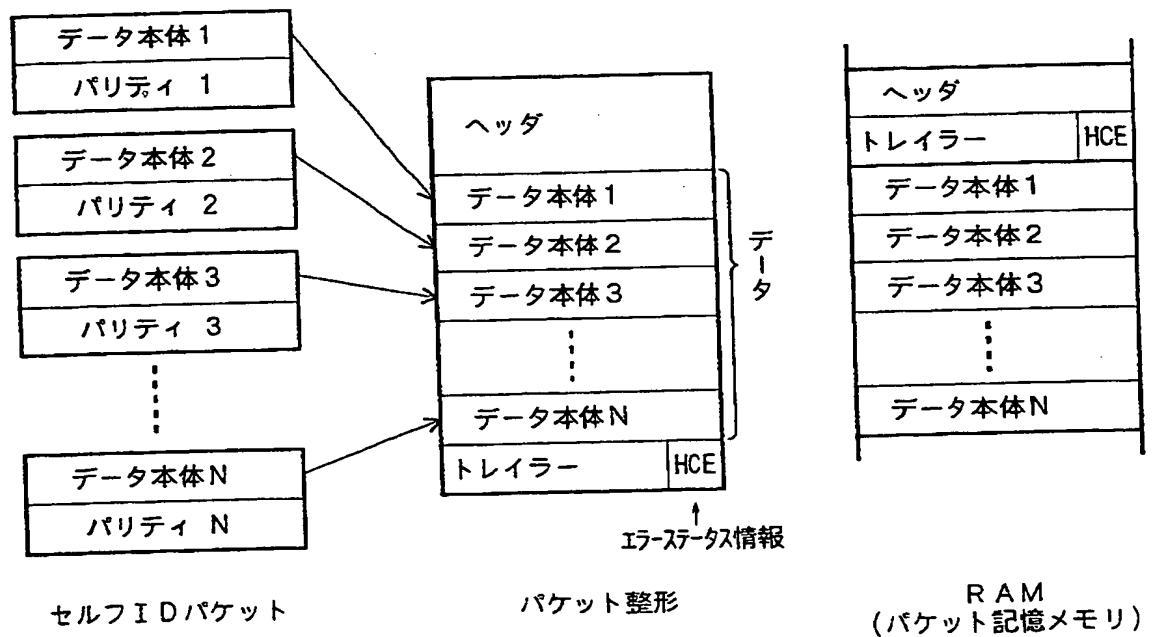


【図 14】

(A)

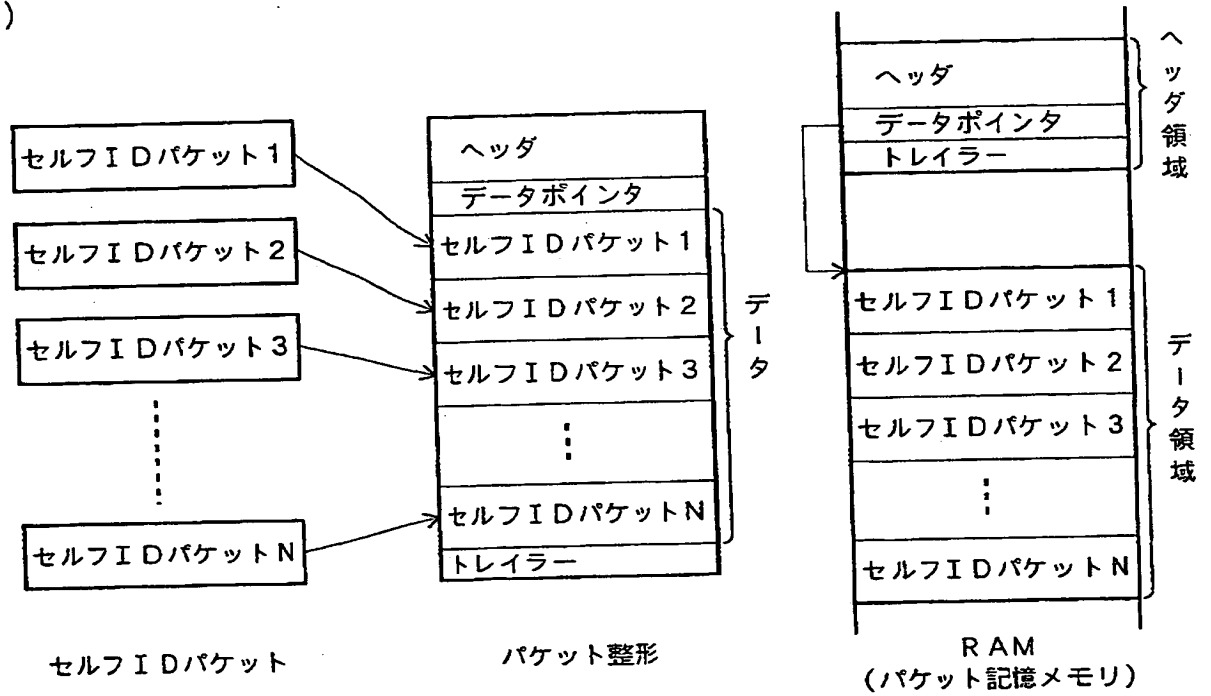


(B)

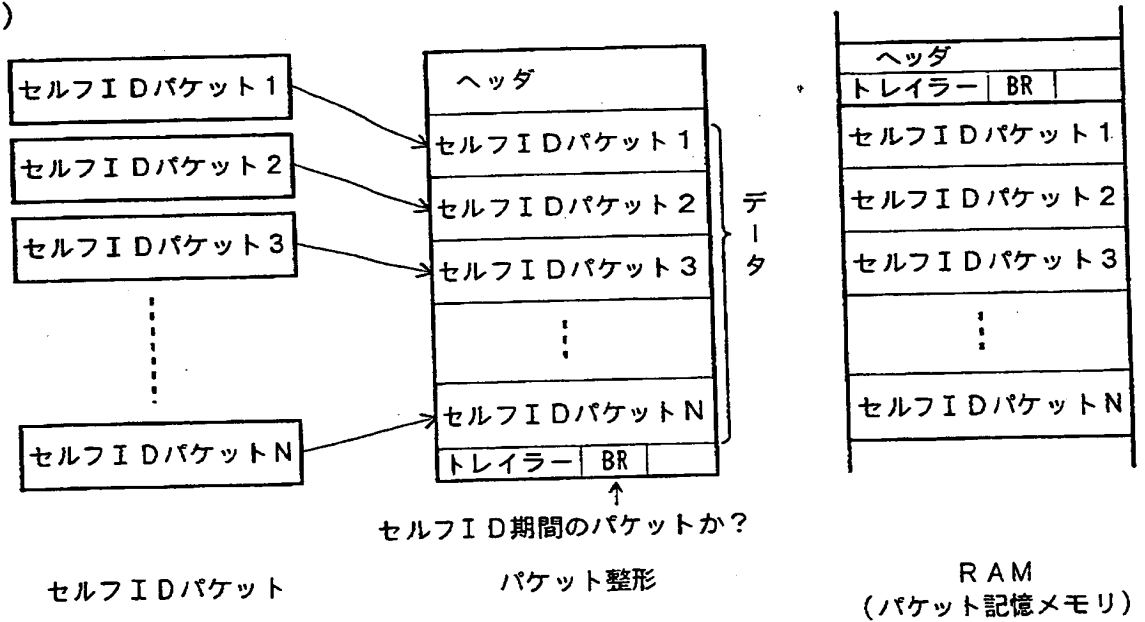


【図15】

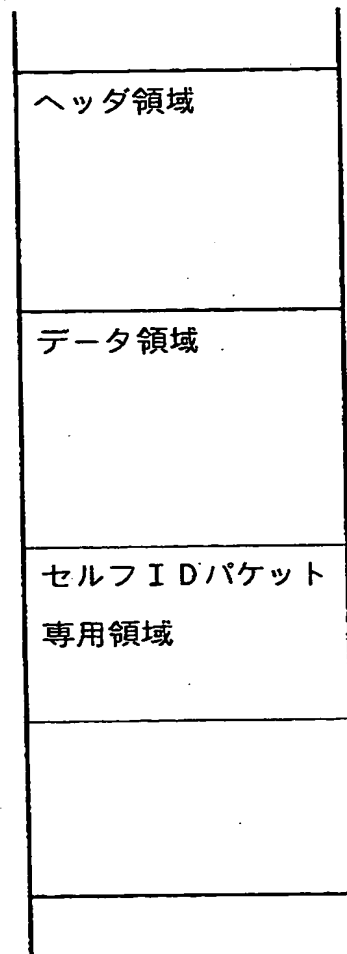
(A)



(B)



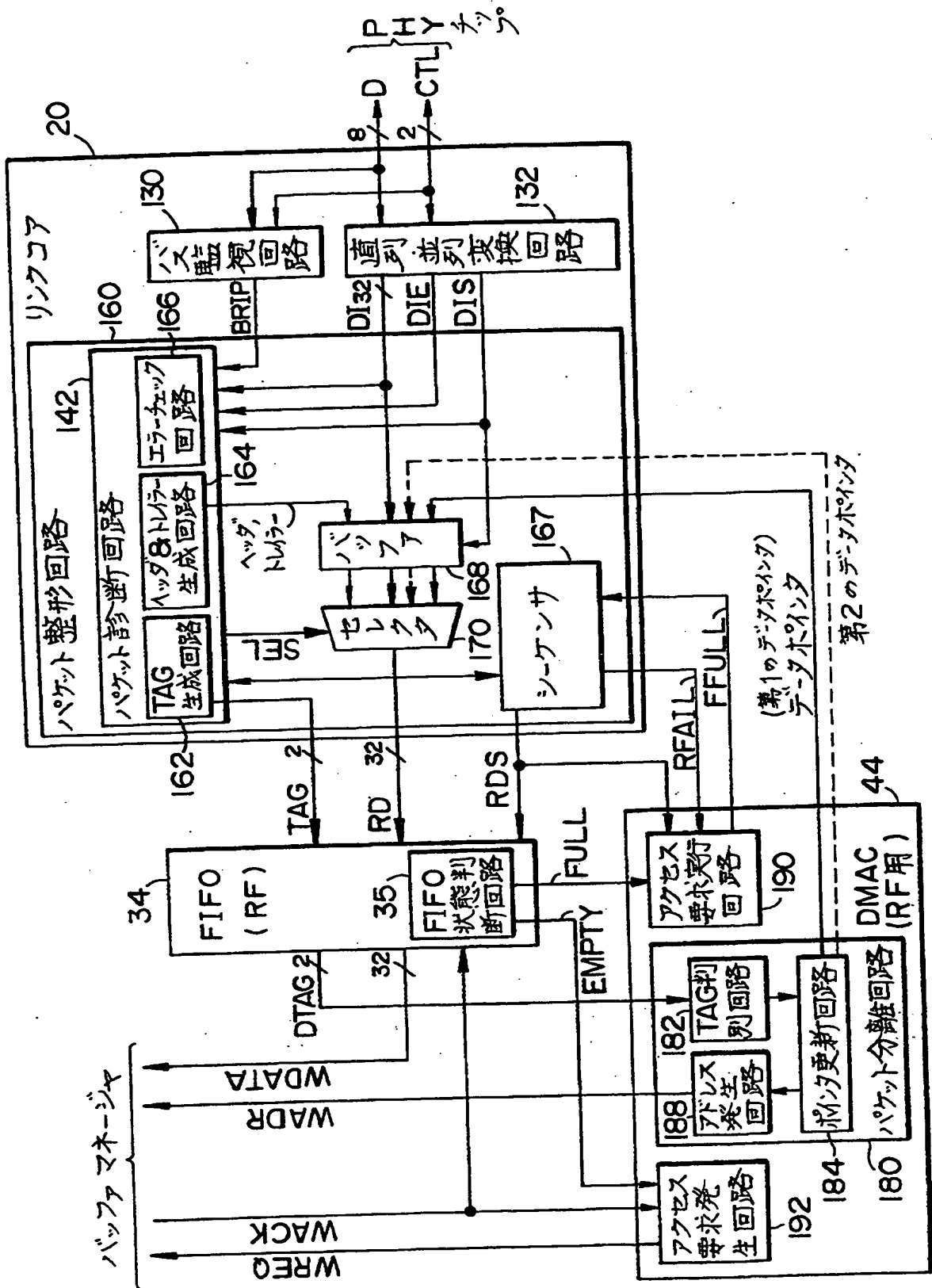
【図 16】



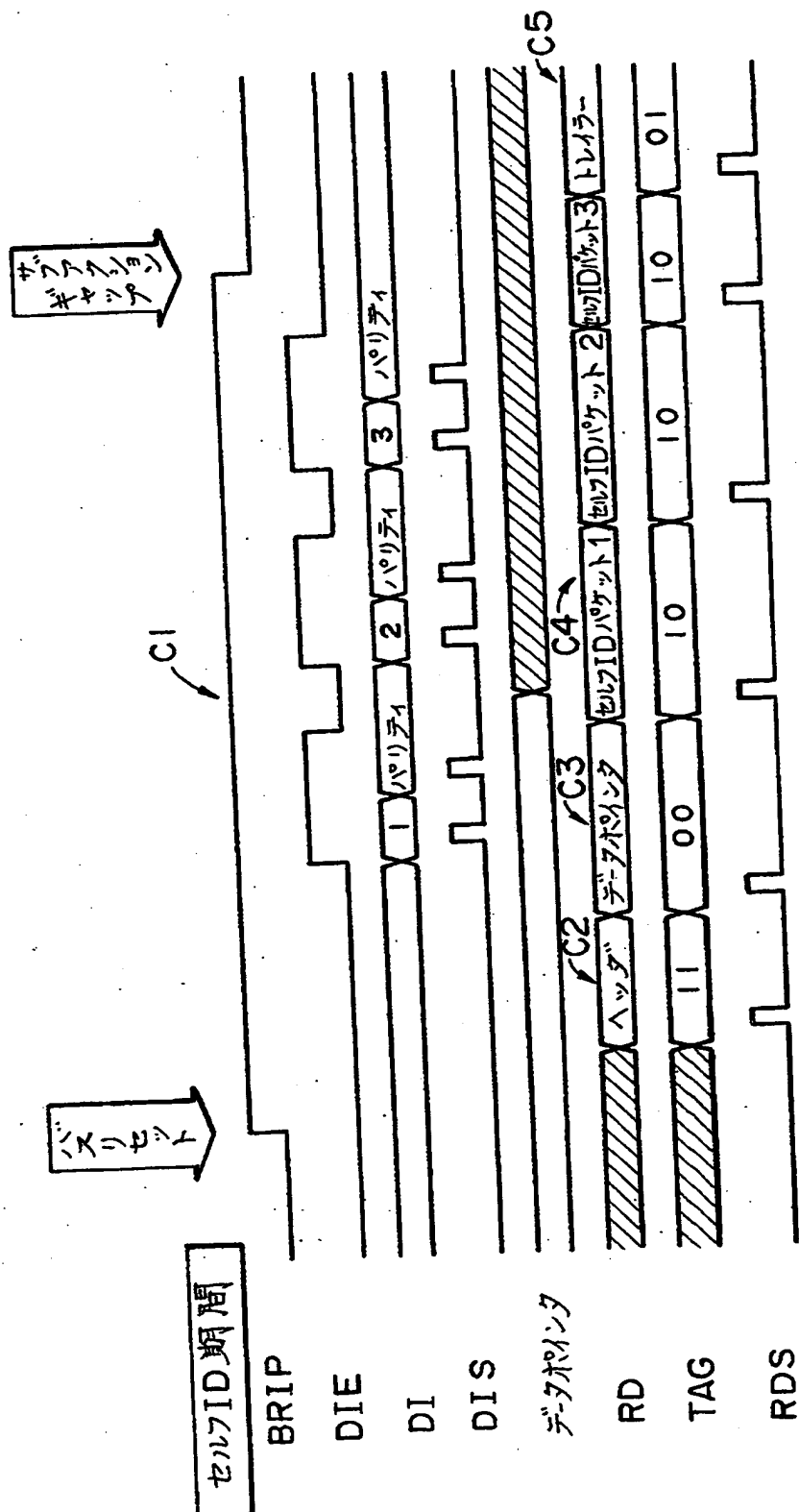
RAM (バケット記憶メモリ)

特平10-321540

【図17】

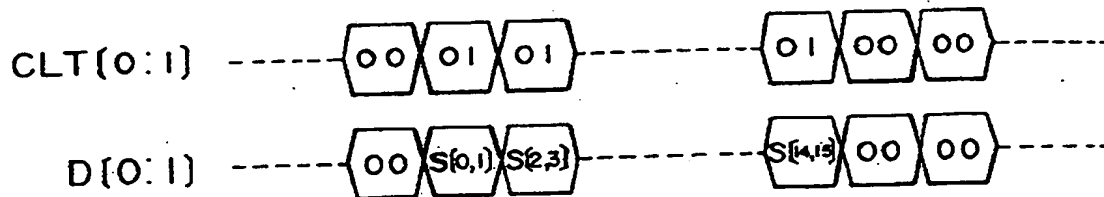


【図 18】



【図19】

(A)



(B)

PHYチップ駆動時

CTL	NAME
00	アイドル
01	ステータス
10	受信
11	転送

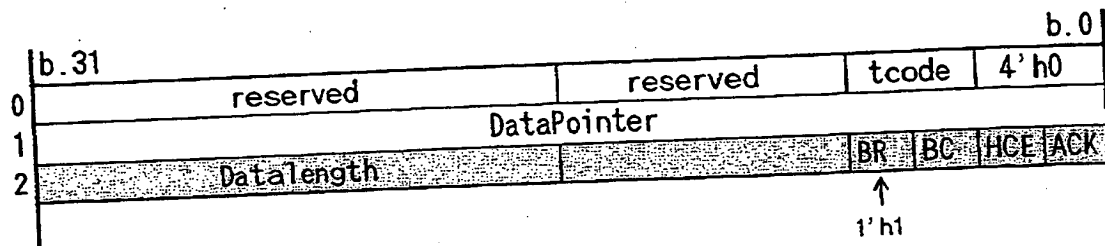
(C)

ステータスビット

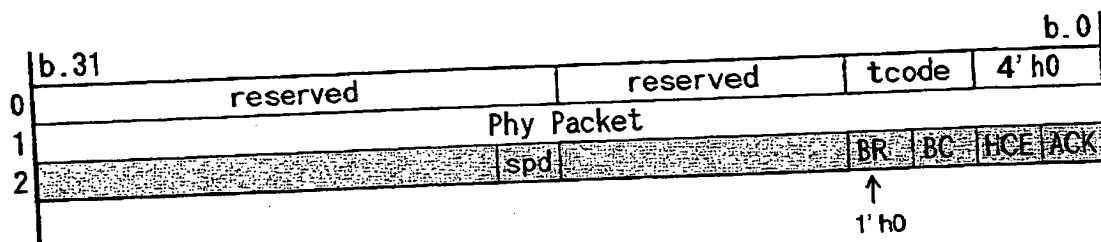
BIT	NAME
0	調停リセットギャップ
1	サブアクションギャップ
2	バスリセット
3	PHY割り込み
4~7	アドレス
8~15	データ

【図 20】

(A) RxAsynchronousBusResetPacket (tcode:0xE)



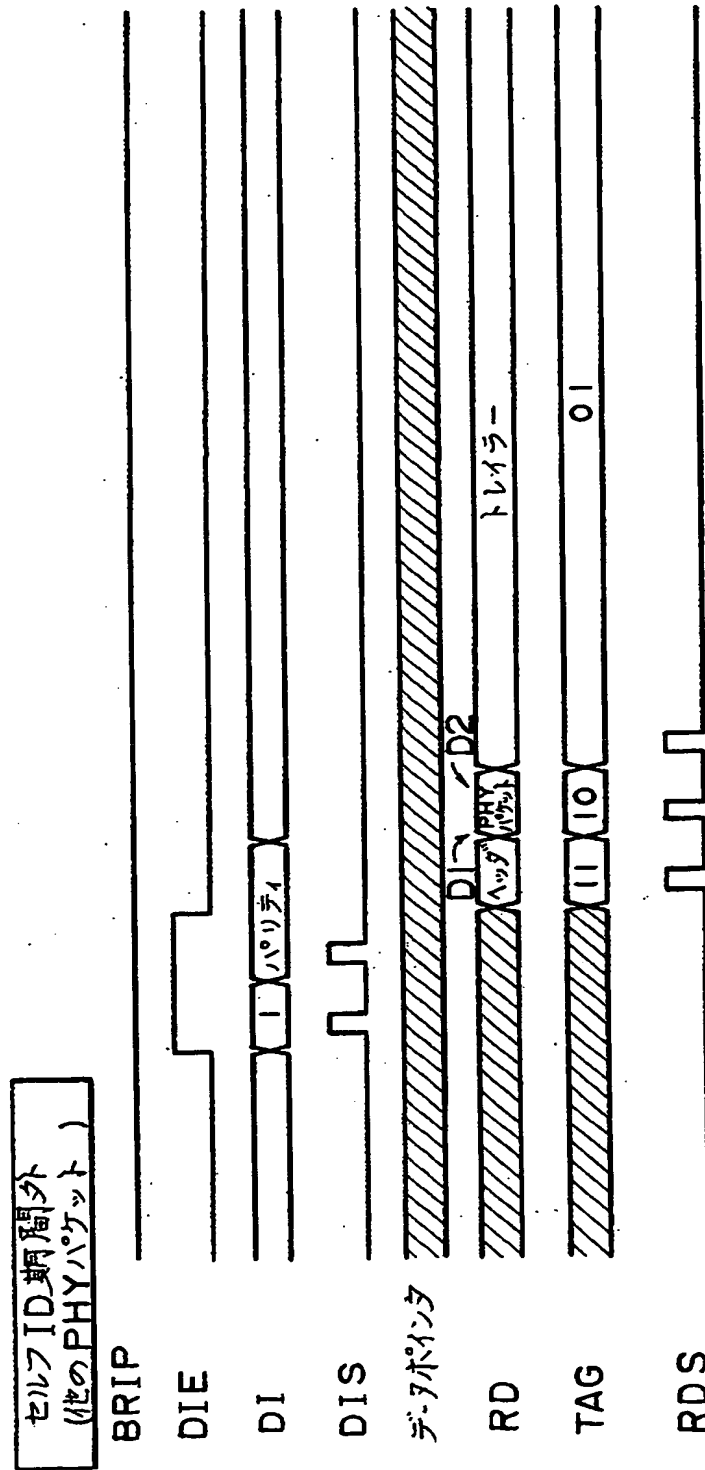
(B) RxAsynchronousPhyPacket Normal (tcode:0xE)



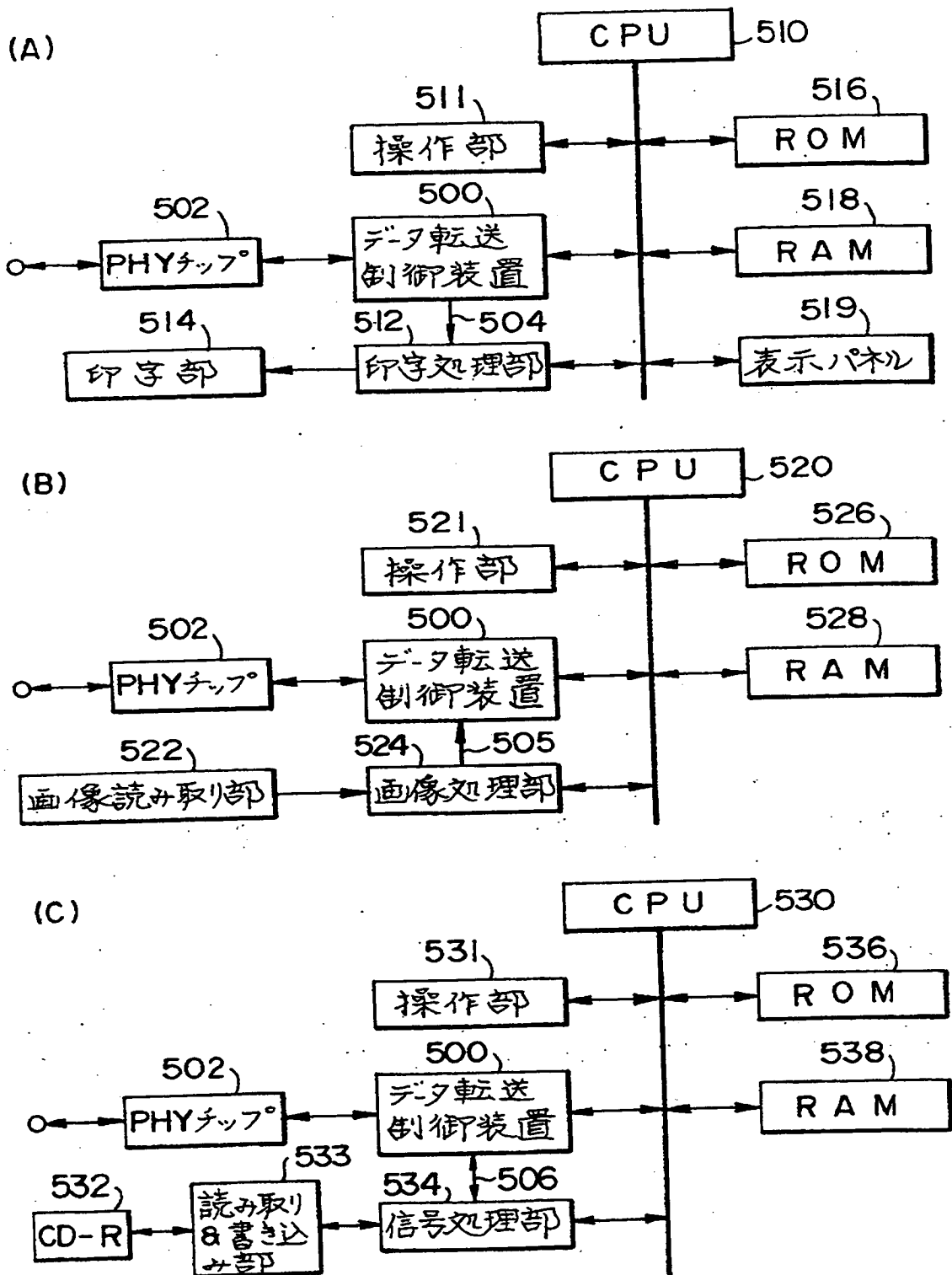
【図 21】

TAG (DTAG)	意 味
00	ヘッダ
01	トレーラー
10	データ
11	スタート

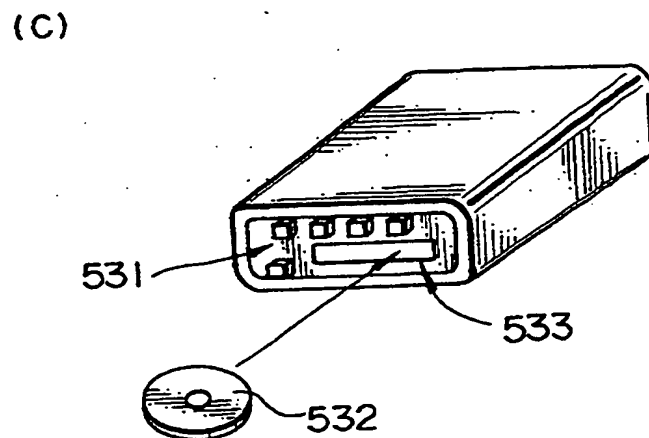
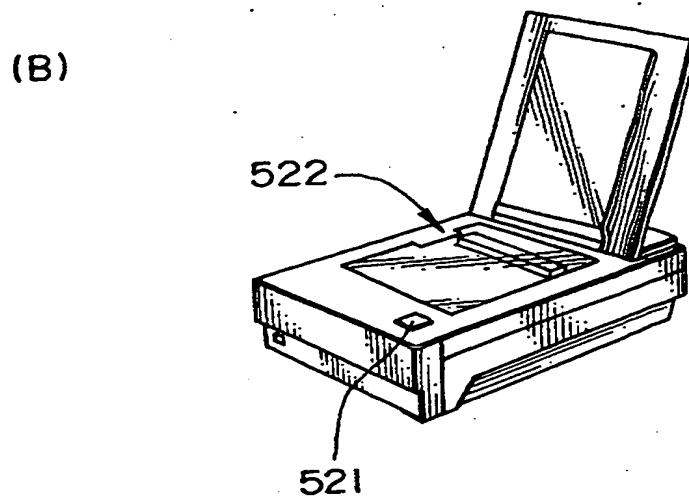
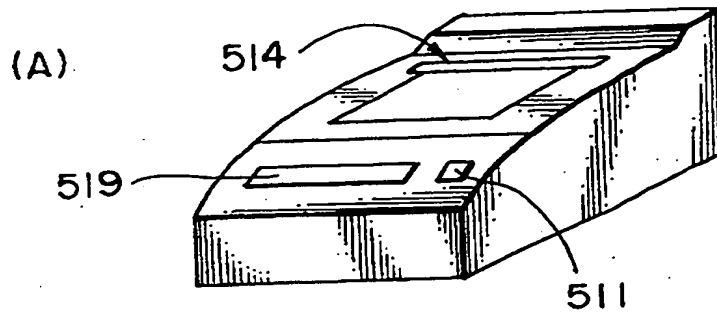
【図 22】



【図 23】



【図 24】



【書類名】 要約書

【要約】

【課題】 各ノードが有するリソースを有効利用でき、処理のオーバーヘッドを軽減できるデータ転送制御装置、電子機器を提供すること。

【解決手段】 パケット整形回路が、各ノードから送られるIEEE1394規格のセルフIDパケットを受け、一連のセルフIDパケットの並びからなるデータとヘッダとによりフレームが構成されるパケットに整形し、上層にインターフェースする。セルフIDパケットのパリティを削除し、パリティが削除されたセルフIDパケットの並びによりパケットのデータを構成すると共に、エラーステータス情報をパリティのトレイラーに付加する。パケットのヘッダをヘッダ領域にデータをデータ領域に分離して書き込むと共に、データのアドレスを示すデータポインタをパケットのヘッダに付加する。データ領域にセルフIDパケット専用の領域を設ける。セルフID期間中か否かを検出し、セルフID期間中に転送されてきたパケットをセルフIDパケットと見なしてパケット整形を行う。

【選択図】 図15

【書類名】 職権訂正データ
 【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】

【識別番号】 000002369

【住所又は居所】 東京都新宿区西新宿2丁目4番1号

【氏名又は名称】 セイコーエプソン株式会社

【代理人】 申請人

【識別番号】 100090479

【住所又は居所】 東京都杉並区荻窪5丁目26番13号 荻窪TMB
 ル2階 井上・布施合同特許事務所

【氏名又は名称】 井上 一

【代理人】 申請人

【識別番号】 100090387

【住所又は居所】 東京都杉並区荻窪5丁目26番13号 荻窪TMB
 ル2階 井上・布施合同特許事務所

【氏名又は名称】 布施 行夫

【代理人】 申請人

【識別番号】 100090398

【住所又は居所】 東京都杉並区荻窪5丁目26番13号 荻窪TMB
 ル2階 井上・布施合同特許事務所

【氏名又は名称】 大淵 美千栄

出 願 人 履 歴 情 報

識別番号

[000002369]

1. 変更年月日	1990年 8月20日
[変更理由]	新規登録
住 所	東京都新宿区西新宿2丁目4番1号
氏 名	セイコーエプソン株式会社